IRE
Transactions
on ELECTRONIC COMPUTERS

Volume EC-10 June, 1961 Number 2

Published Quarterly

TABLE OF CONTENTS

Contributions

Correspondence

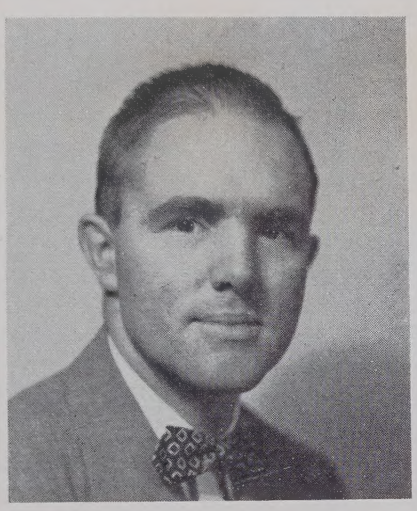
PUBLISHED BY THE
Professional Group on ELECTRONIC COMPUTERS

IRE Transactions® on Electronic Computers

Published by The Institute of Radio Engineers, Inc., for the Professional Group on Electronic Computers at 1 East 79 Street, New York 21, N.Y. Responsibility for the contents rests upon the authors and not upon the IRE, the Group, or its members. Individual copies of this issue and all available back issues may be purchased at the following prices: IRE members (one copy) $2.25, libraries and colleges $3.25, all others $4.50. Annual subscription price: libraries and colleges $12.75; non-members $17.00. Address requests to the Institute of Radio Engineers, 1 East 79 Street, N.Y. 21, N.Y.

Notice to Authors: Address all papers and editorial correspondence to the appropriate Editor. Addresses are listed on page 343. To avoid delay and inconvenience in the processing of manuscripts, please follow the procedure suggested there.

New Editor-in-Chief of IRETEC



Norman R. Scott

Norman R. Scott (SM'46) was born in Brooklyn, N. Y., on May 15, 1918. He received the S.B. and S.M. degrees from the Massachusetts Institute of Technology, Cambridge, Mass., in 1941, and the Ph.D. degree in Electrical Engineering from the University of Illinois, Urbana, Ill., in 1950.

His professional experience includes a year with General Electric as an M.I.T. coöp student, followed by five years in the U. S. Army during World War II, spent mostly at Wright Field, where he served ultimately with the rank of Major in the position of Chief of the Special Projects Laboratory. He was a faculty member at the University of Illinois during the next four years.

Since 1951 he has been on the faculty of the University of Michigan, Ann Arbor, Mich., where he is now, as Professor of Electrical Engineering, in charge of the Digital Computer Engineering Laboratory and courses in this area.

Dr. Scott is the author of several papers and articles, as well as the book, "Analog and Digital Computer Technology," recently published by McGraw-Hill. He was one of four American delegates who visited Soviet computer laboratories in 1958. He is currently a member of the AIEE Computing Devices Committee, and of Sigma Xi, Eta Kappa Nu, and Phi Kappa Phi. He has served on the Administrative Committee of the IRE-PGEC and as Member-at-Large for Engineering on the Council of the Association for Computing Machinery. He also acts as a consultant to several firms on problems of computer engineering.

Editorial

Our very best wishes for every success go to the new Editor-in-Chief of the TRANSACTIONS, Dr. Norman R. Scott, Professor of Electrical Engineering at the University of Michigan. We hope that he will get as much enjoyment from this assignment as we have. It has been a very exciting and interesting two and one-half years, during which we have processed over four hundred manuscripts, corresponded with about that number of authors, and made many new friends. A career change makes our retirement from the Editorship necessary.

At this time we also want to express our sincere thanks to the large and able group of referees who have read and evaluated contributed papers. The computer field is far too large and diverse to be encompassed effectively by one man. Hence, the Editor has relied heavily on the skill and judgment of the referees. The proof of their judgment lies in the caliber of the TRANSACTIONS, for the Editor is primarily the vehicle through which they speak.

Even with the help of able referees, the IRETEC Editor's job had grown too large to be carried effectively by a single individual as a venture supplementary to his principal occupation. Of the alternative methods of lightening the load, the most suitable has seemed the creation of Associate Editors for specific areas of the field.

We have just begun the process of creating Associate Editors. The first was installed in April, when John E. Sherman of Lockheed became Associate Editor for Analog and Hybrid Computers. All manuscripts in that area for publication in regular issues of the TRANSACTIONS will be processed through Jack Sherman's office, and may be sent to him directly. (Mr. Sherman's photograph and biography appear in the PGEC News Section on page 340.)

Not to confuse things too much, the February, 1962, issue of IRE TRANSACTIONS ON ELECTRONIC COMPUTERS will be a special one on Analog and Hybrid Computers, and John McLeod of Convair is Guest Editor. Papers for that issue—particularly survey and state-of-the-art papers—should be sent directly to John McLeod (see PGEC News, page 338, for his photo and biography).

Incidentally, the February, 1962, special issue will mark the beginning of bimonthly publication for IRE TRANSACTIONS ON ELECTRONIC COMPUTERS, a step now overdue because of the large number of good manuscripts being received. Less delay in publication will result under the bimonthly scheme.

Norm Scott is taking over at a particularly significant point in the history of the computer field. It is now evident that the computer will profoundly affect the economic and political systems of the world as well as its science and technology. Like it or not, we have opened yet another of Pandora's boxes, and no one that we know of has good answers to the many new problems created.

In this situation the role played by professional journals is to provide clear and unbiased reporting of the state of the art. We look forward to a highly successful carrying out of that task by IRE TRANSACTIONS ON ELECTRONIC COMPUTERS under Norman Scott's Editorship. Please support him as you have us, by advice, service, and the contribution of manuscripts.

HOWARD E. TOMPKINS
105 Russell Street
Ridley Park, Pa.

A Straightforward Way of Generating All Boolean Functions of N Variables Using a Single Magnetic Circuit*

K. V. MINA† AND E. E. NEWHALL†, ASSOCIATE MEMBER, IRE

Summary—A correspondence has been established[1] between the topology of relay contact networks and the topology of magnetic circuits, which may be applied to a relay tree to produce a magnetic structure capable of generating, in a simple manner, all Boolean functions of N variables. Once the basic magnetic topology is established, it may be distorted to achieve winding simplicity at the expense of magnetic circuit complexity. In the resulting arrangement, drive, hold (variable) and reset windings are always in the same position, regardless of the function to be generated. Any one of the functions of N variables is generated by linking a selected group of the output legs.

The structure is such that all switching paths are of equal length, causing all outputs to be equal in amplitude. This balanced arrangement also permits the holding MMF to be significantly smaller than the drive MMF. The holding scheme is a symmetrical one, specifically arranged to overcome shuttle flux problems and reduce noise. An 8-leg manganese magnesium zinc ferrite structure is operated easily at a 4-μsec cycle time with an output of 500 mv into 5 ohms. The peak-signal-to-peak noise ratio was at worst 8:1. A 1-in-256 selector, using seventeen 16-leg structures, is under construction. The structure described here is in a sense the complement of the laddic,[2] in that the drive and hold windings are always applied in fixed positions and different functions are generated by linking different sets of output legs.

I. Introduction

A CORRESPONDENCE has been established[1] between the topology of relay contact networks and the topology of magnetic circuits, which may be applied to a relay tree to produce a magnetic structure capable of generating, in a simple manner, all Boolean functions of N variables. Logically, we are then in the following position. Rather than compose logical functions by combining magnetic circuits capable of performing AND, OR, etc., we generate all Boolean functions of N variables simultaneously on one magnetic circuit. For example, a structure with 4 legs will generate simultaneously all Boolean functions of 2 variables and a structure with 8 legs will generate simultaneously all Boolean functions of 3 variables, etc. In the resulting structure, the drive, hold and reset windings are always in the same position, regardless of the function or functions to be generated. Any of the 2^{2^N} functions of N variables is generated by linking a selected group of output legs. For example, a single structure with 8 legs and fixed drive (one location only), reset and hold windings can simultaneously generate all 256 functions of 3 variables, using one output winding per function.

The structure is such that all switching paths are of equal length, causing all outputs to be equal in amplitude. This balanced arrangement permits the holding MMF to be significantly smaller than the drive MMF. The interconnecting legs are initially neutral, and if a leg is held, the neutrality of the leg is maintained. The holding scheme is a symmetrical one, specifically arranged to overcome shuttle flux problems and to reduce noise.

An 8-leg manganese magnesium zinc ferrite structure of average cross section 100×130 mils and a mean path length of 3 inches required a driving MMF of 2 ampere turns (concentrated at one location), a hold MMF of a $\frac{3}{4}$ ampere turn, and operated easily at a 4-μsec cycle time with an output of 500 mv into 5 ohms. The peak-signal-to-peak-noise ratio was at worst 8:1. A 16-leg structure was operated in a similar fashion. A 1-in-256 selector using seventeen 16-leg structures is under construction.

From a logical point of view the structure is appealing in the sense that only one type of wired structure is required for realizing a multiplicity of functions simultaneously.

II. Operation

Fig. 1 shows the magnetic analog of a relay contact.[1] If the reset pattern is as shown, and we attempt to switch flux down the leg shown in Fig. 1(a), then it is impossible for this flux to close through the right side of the leg because the leg is already saturated down on this side. Also, if at the same time we try to close the flux through this leg and we hold the left side up by passing current through the hold windings shown, then the leg is blocked completely as far as the transmission of flux is concerned. Fig. 1(b) is a modified version of this switch organized so that shuttle flux is confined to the held leg. Experimentally, it has been observed that if one has a number of legs in parallel of the type shown in Fig. 1(a), the accumulation of shuttle flux due to super saturation by the hold current will sometimes be great enough to completely switch an unheld leg. On the other hand, if one organizes things as shown in Fig. 1(b), then shuttle

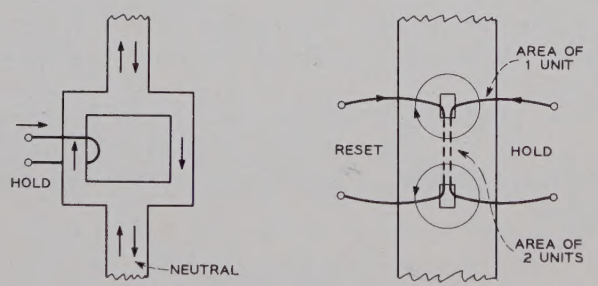
* Received by the PGEC, November 15, 1960; revised manuscript received, February 10, 1961. Presented at the 7th Annual Denver Res. Inst. Symp. on Computers and Data Processing, Estes Park, Colo.; July 28–29, 1960.

† Bell Telephone Labs., Inc., Murray Hill, N. J.

[1] D. B. Armstrong, T. H. Crowley, U. F. Gianola, and E. E. Newhall, "Magnetic analogs of relay contact networks for logic," IRE TRANS. ON ELECTRONIC COMPUTERS, pp. 30–35; March, 1960.

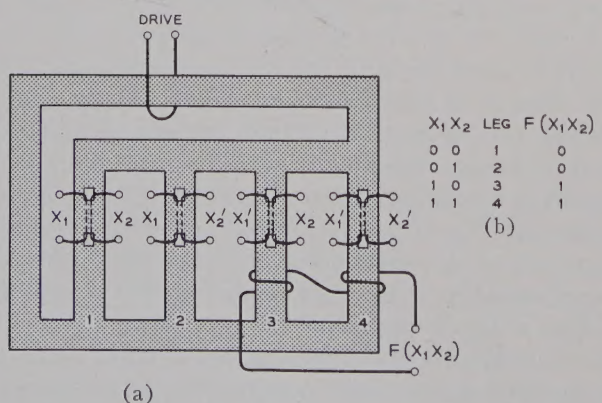
[2] U. F. Gianola and T. H. Crowley, "The laddic," *Bell Sys. Tech. J.*, vol. 38, pp. 45–72; January, 1959.

flux developed by excessive hold current will merely close around the two holes shown and will remain confined to the legs being held. If the reset pattern is as shown, and we attempt to switch flux down the leg shown in Fig. 1(b), it cannot go down the right side of the top hole because this is already saturated down, and it cannot go down the left side of the bottom hole because it also is saturated down. Furthermore, if the hold winding shown is energized, it cannot switch the left side of the top hole nor can it switch the right side of the bottom hole; thus the leg is held. Consider placing a number of these switches in the structure shown in Fig. 2. Logically, this structure is just the magnetic analog of a relay tree. Initially, a reset applied to all the holes closes flux around the holes, as described previously. The interconnecting legs are then neutral and upon simultaneous application of drive and hold currents, the structure will behave as follows. Suppose the variables are wound as shown, and suppose $X_1=1$ and $X_2=0$, then legs 1 and 2 will be held by X_1 and legs 2 and 4 will be held by X_2'. Thus only leg 3 will switch. An examination of the structure reveals that under normal operating conditions, one and only one leg will switch corresponding to the particular input code which has been applied. The input codes, together with the associated legs, are shown in Fig. 2(b). The way in which one would generate a function is clear from examination of Fig. 2(b). Suppose we wish to generate the arbitrary function shown. Then when either leg 3 or 4 switches this function should be a 1; and so by placing a series



(a) (b)

Fig. 1—Magnetic analog of a relay contact.



(a)

Fig. 2—Magnetic analog of a relay tree adjusted so that all switching paths are the same length.

winding which links legs 3 and 4, as shown in Fig. 2(a), this function will be developed. It is apparent that any function of two variables is available on this structure and any or all of the 16 functions of two variables may be taken out simultaneously.

Note that the hold windings are placed on the structure so that one variable is associated with exactly half the output legs and the complement of that variable is associated with the remaining half. By extending this winding principle to a 3-dimensional array, 4-leg structures can be stacked to make a 4-variable logic structure The manner of stacking is shown in Fig. 3. Here we see the top sectional view of five 4-output-leg structures; one structure driving four others through the electrical couplings shown. The 4-output structures are lined up so that the wires carrying the holding or variable currents can be passed directly through all 4 structures. Thus 2 of the binary variables select one leg in each of the 4 structures, and the other 2 binary variables select one leg in the driving structure. Upon simultaneous application of drive and holds one of the 4-output structures is driven, and one leg in that structure is selected so as to obtain 1-in-16 selection. Experimental work is being carried on which is aimed at constructing a 1-in-256 selector which is obtained by stacking sixteen 16-leg structures and driving these structures from one 16-leg structure.

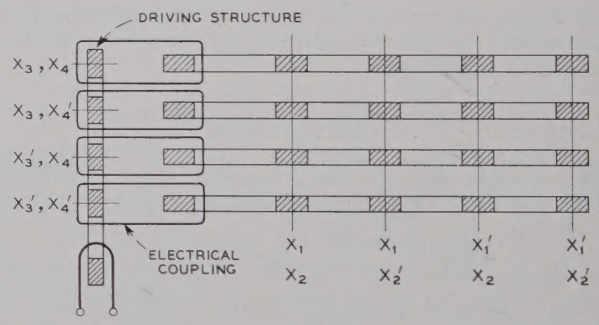


Fig. 3—Stacked logic elements.

The structure shown in Fig. 2 may be controlled with an amount of hold MMF which is small compared to the drive MMF required. No matter which leg is selected, the total switching path length is the same, thus all the output pulses are equal in amplitude. When one starts with this *balanced arrangement* a small amount of hold MMF may be used to upset the balance causing the one unheld leg to be switched. Furthermore, all hold MMF's need only exceed some minimum value and, due to the symmetrical holding arrangement, excessive holding does not result in increased noise [see Fig. 1(b)]. A first estimate of the required hold current was obtained from considerations of the static hysteresis loop alone. To understand this, consider Fig. 4. In this figure we show a detailed view of two legs, side by side, and consider one to be held and the other to be unheld. We seek to predict the minimum value of hold current required.

Suppose we have the B-H loop of the material, and we know the initial condition of legs A and B to be as shown in Fig. 1(b). Then we can find the minimum value of MMF which is just sufficient to start switching leg A, and the maximum value of MMF which is required to just finish switching leg B. The MMF which exists across leg B, when it just finishes switching is $H_{max}l_1$. Here H_{max} is taken from the B-H loop of a thin-walled toroid as shown in Fig. 5. Also, the minimum MMF required to just start switching leg A is $H_{min}l_1$. Thus the hold MMF required, so that leg A just starts switching, is given by the following relationship:

$$H_{max}l_1 - H_{min}l_1 = 0.4\pi NI \qquad (1)$$

where I=amps, H=oersteds, l_1= cm, and NI is the number of hold ampere turns.

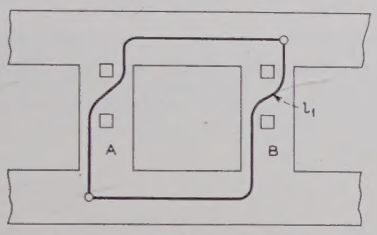


Fig. 4—Paths used to determine the minimum holding current.

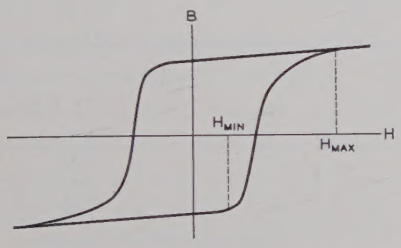


Fig. 5—B-H loop of a thin-walled toroid showing H_{max} and H_{min}.

The simple analysis outlined above gives a lower bound on the required hold current, valid for slow speed operation. This model requires modification as the switching speed is increased to take into account widening of the hysteresis loop due to damping of the domain-wall motion. However, the simple reasoning outlined above does give some basis for the experimental observation that the required holding MMF is considerably smaller than the drive MMF. A little reflection shows that if $H_{max}=H_{min}$, then, in the low-speed case, the required holding current would approach zero.

III. Description of the Experimental Structure

A 4-, 8-, and 16-leg structure made from manganese magnesium zinc ferrite are pictured in Fig. 6. These structures are capable of generating all Boolean functions of 2, 3, and 4 variables, respectively.

The drive, reset, and hold windings are placed on these structures in a manner similar to that discussed earlier. The 16-leg structure and its winding configuration will be considered more fully.

Fig. 7 shows a 16-leg structure. The drive rail is flux limited so that it can carry an amount of flux at most equal to the flux which an output leg can carry. Note also that all switching paths are the same length. The particular dimensions of the structure were determined by the output power requirements.

The 4 variables and their complements are threaded through the small holes in each rung in a specified

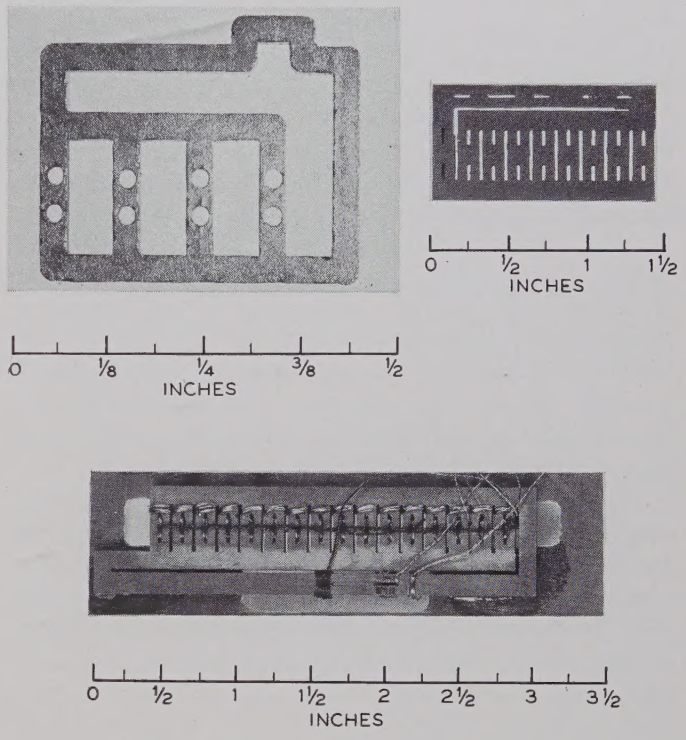


Fig. 6—4-, 8- and 16-leg logic structures.

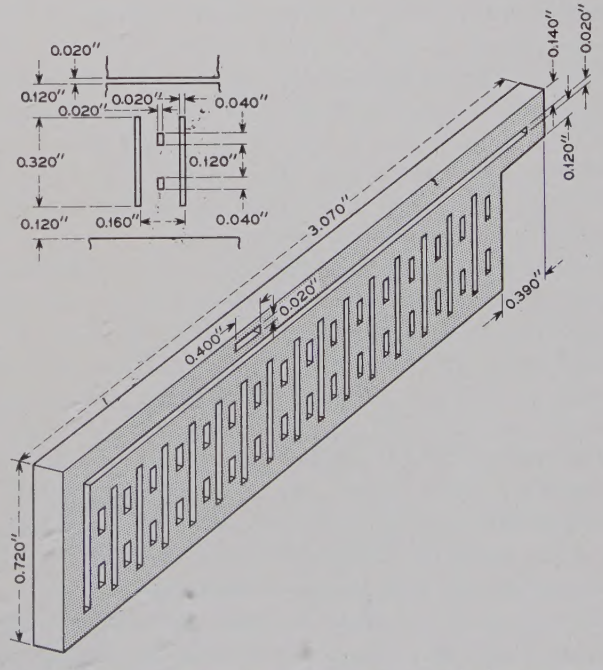


Fig. 7—Geometry and dimension of a 16-leg logic structure.

manner. Many winding configurations are possible, subject to the restriction that in any allowed winding configuration each variable combination results in a different leg not held. This 16-leg structure is wound with a "binary" configuration, where, as the input codes vary from binary 0 through binary 15, the leg selected progresses from left to right successively down the structure. This winding configuration is shown symbolically in Fig. 8.

The structure is reset to the proper neutral condition by two windings, reset-A(R_A) and reset-B(R_B) (Fig. 8). These two windings are activated simultaneously. The reset-A winding provides a larger MMF than reset-B and neutralizes the drive rail by switching flux around the large outer path. This winding is applied at a single physical location on the structure, so that a multiturn winding is practical. The reset-B winding is distributed amongst all the legs to insure that each leg individually returns to the proper neutral pattern. The reset-B need only guide the reset-A MMF such as to cause switching through the proper leg, *i.e.*, reset the leg just selected. Note the structure may be operated with only a reset-B winding. In this case, however, the reset-B winding must supply enough MMF to switch around the large path.

The structure is driven on the drive rail by a drive winding shown in Fig. 8. This must supply substantial MMF since it must switch around the larger outer path, but, as with the reset-A winding, multiple turns can be used. It should be noted that either direction of drive current can be used without reversing either the variable or reset directions.

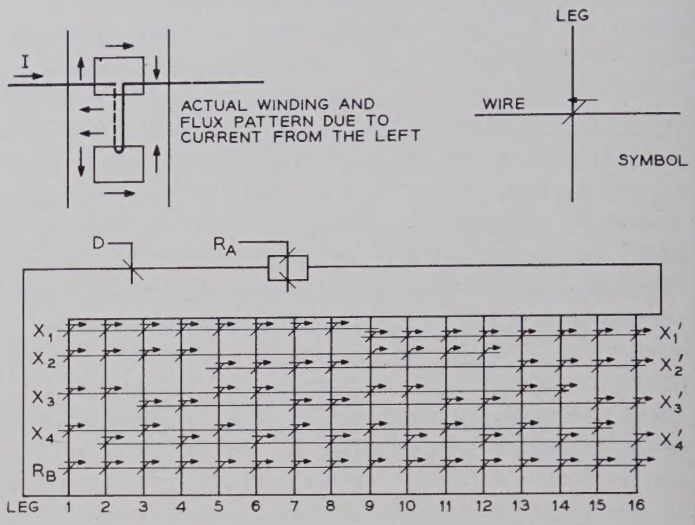


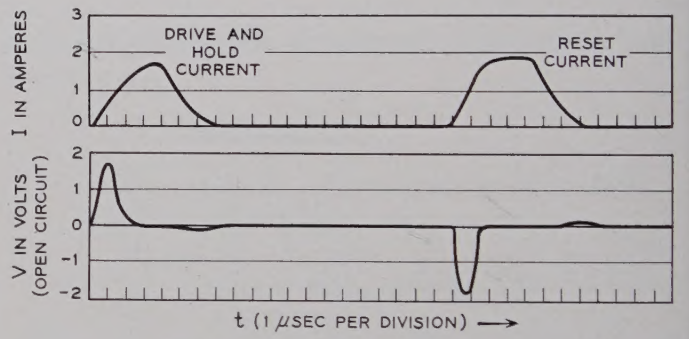
Fig. 8—Wiring details of a 16-leg logic structure. With this winding symbolism, if the current is diverted by the winding symbol towards the arrow, the flux between the two small holes in a leg is switched in the direction of the arrow; while if current is diverted away from the arrow, flux switches in the direction opposite to the arrow.

IV. Experimental Results

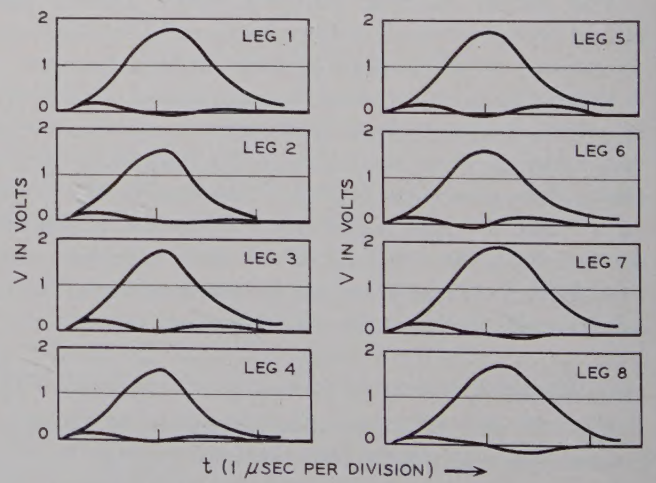
The 8-leg structure shown in Fig. 6 was wound with a 5-turn drive winding, a 2-turn hold winding, a 2-turn reset-B winding, and a 2-turn output winding. (This structure was operated with a reset-B winding only.) The drive, hold and reset currents in these windings are shown in Fig. 9(a), together with a typical open-circuit switching waveform. Fig. 9(b) shows the switching waveform and worst noise condition on each of the 8 legs using an enlarged time scale.

During the 2.2-μsec switching time associated with the drive phase, the drive current is essentially a ramp rising at the rate of 0.6 ampere per μsec. This supplies a ramp MMF rising at the rate of 3 ampere-turns per μsec. During the 1.5-μsec switching time associated with the reset phase, the reset current is essentially a ramp rising at the rate of 1.2 ampere per μsec. This supplies a ramp MMF of 2.4 ampere-turns per μsec. The hold current was the same as the drive current, so that during the drive phase the hold MMF was essentially a ramp rising at the rate of 1.2 ampere-turns per μsec.

During the drive phase the peak open-circuit output voltage from the 2-turn output winding varied from



(a)



(b)

Fig. 9—Waveforms associated with an 8-leg structure.

1.60 to 1.90 volts. The peak-signal-to-peak-noise (in the positive direction) was greater than 8:1 in all cases. The peak output voltage was reduced to half its value by a 5-ohm load.

The 16-leg structure was wound with a 15-turn drive winding, a 2-turn hold winding, effectively a 10-turn reset-A winding, a 2-turn reset-B winding, and 4-turn output windings. The drive hold and reset currents are shown in Fig. 10(a) along with a typical loaded switching waveform. The open-circuit switching waveforms on the 16 legs are shown in Fig. 10(b). Note that the time scales differ in Fig. 10(a) and (b).

During the 5.0-μsec switching time associated with the drive phase, the drive MMF is approximately a ramp rising at the rate of 3 ampere-turns per microsecond for 3 μsec, followed by a decay at a rate of 2.5 ampere turns per μsec. The variable MMF is constant at the value of 1 ampere-turn. The peak open-circuit output voltage in 4 turns varied over the 16 rungs, from 1.32 to 1.45 volts. The peak-signal-to-peak-noise (in the positive sense) is greater than 9:1 in all cases. During the 4-μsec switching time associated with the reset phase, the reset-A and -B MMF's are ramps rising at 1.8 and 0.36 ampere-turns per μsec for 3.5 μsec, followed by decays at the rate of 1.4 and 0.28 ampere-turns per μsec.

The waveforms shown in Figs. 9 and 10 indicate that:

1) Due to the equal path length feature, the output voltages are approximately equal.
2) Due to the symmetrical holding method, the noise due to the hold currents is small.
3) Due to the particular geometrical arrangement and winding arrangement of the structure, all hold MMF's can be equal, and there exists a significant MMF gain from drive to hold.

Finally, consider a simple experiment which shows how flux divides between 2 legs, as a function of the hold MMF applied to one leg. An analysis of these results suggests some of the factors which must be included in a model which is to accurately represent the structure.

V. Experimental Investigation of Hold MMF Required

The 16-leg structure was operated with a permanent hold on 14 of the legs and an adjustable hold on one of the legs and no hold on the other leg. For this experiment, leg 16 was held with the adjustable hold and leg 8 was not held. The drive and hold currents were applied as described earlier. The reset current was applied as a slowly rising ramp. During the slow reset state, a flux-MMF loop tracer was used to find the flux condition of legs 8 and 16. The condition of these legs was plotted as a function of the hold MMF applied to leg 16 during the drive phase. The results of this experiment are shown in Fig. 11. Several peculiarities of the experimental

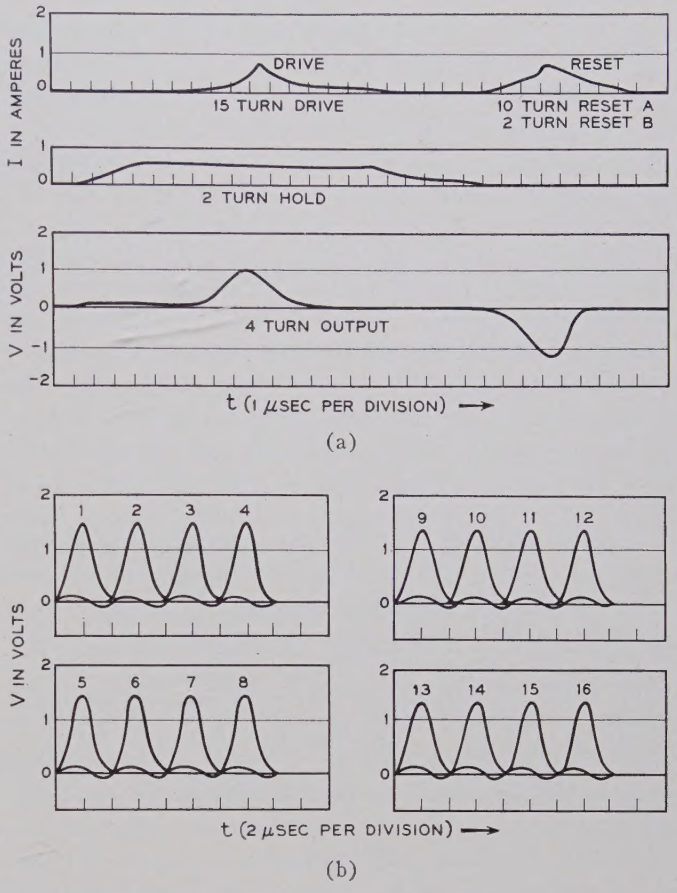


Fig. 10—Waveforms associated with a 16-leg structure. (a) Output voltage into 10 ohms. (b) Output voltage is for an open circuit.

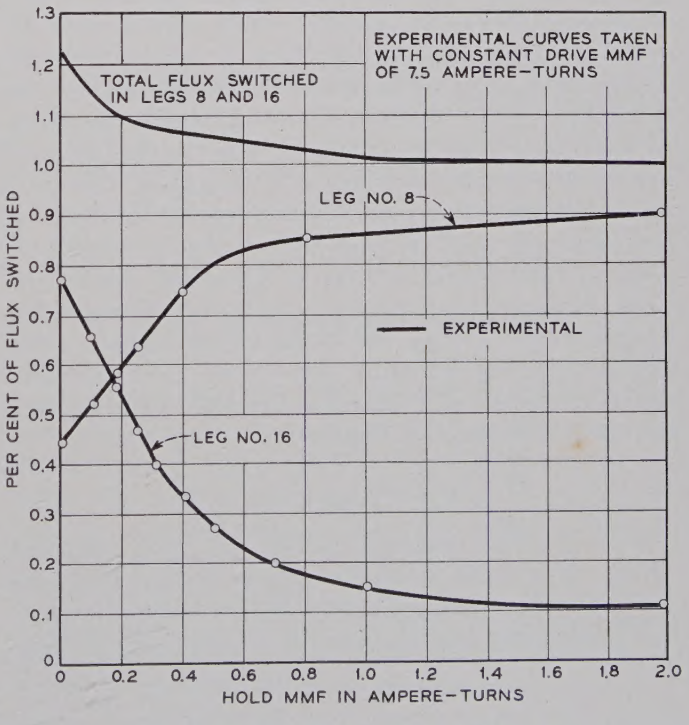


Fig. 11—Holding current required for 16-leg structure.

curve deserve comment. In the experiment the sum of the fluxes which entered legs 8 and 16 depended on the amplitude of the hold current, even though the drive was flux limited and the flux switched by the drive current remained practically constant. This is believed due to an increase in leakage as the hold current increased. The reference value of this sum (100 per cent in Fig. 11) was taken to be the value for large hold currents.

Note also that with zero hold current flux did not split equally between legs 8 and 16. This also may be attributed to leakage. If legs 1–7 were not present, the symmetry of the structure would suggest that legs 8 and 16 should carry the same amount of flux when the hold current is zero. However legs 1–7 remove this symmetry and act as a shunt path, reducing the amount of flux which goes through leg 8, even though legs 1–7 are held.

Finally, observe that the flux through leg 16 approaches a value different from zero as the hold current is increased. This is believed to be mainly due to leakage through leg 16 even though it is "fully held."

A first estimate of the minimum value of the required hold MMF may be obtained using the methods discussed previously. Using H_{max} and H_{min} as shown in Fig. 12, and length l, from Fig. 7, the minimum value of hold MMF was found to be a 0.55-ampere turn. Comparison with Fig. 11 indicates that this could be considered to be a reasonable first estimate. A more detailed calculation can be made, based on the static hysteresis loop, and neglecting leakage. This does not seem warranted because such a model will not predict the leakage effects, discussed above, which appear to appreciably affect the experimental results. In setting up an accurate model to represent the structure, a good low-speed model which accounts for leakage effects should first be established. Once this model is established it could be adjusted to account for the damping of the domain-wall motion associated with higher switching rates. Theoretical and experimental calculations suggest that domain-wall damping may be accounted for by a coupled resistance which varies directly as the area of the leg enclosed and inversely as the length of the leg. This coupled resistance will change the H field distribution in the structure, during switching, in a manner determined by the speed, but the high-speed model will degenerate into the low-speed model, as required.

VII. Conclusions

A single-sheet magnetic circuit has been described which can generate simultaneously all Boolean functions of N variables where N in practice has been as large as 4. The elements may be cascaded to generate functions

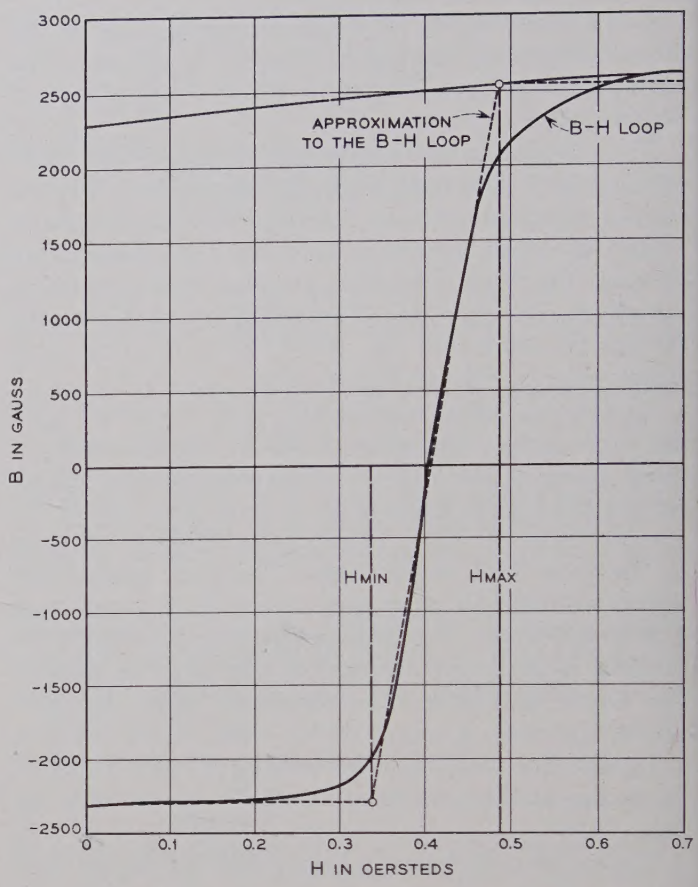


Fig. 12—Static B-H loop for manganese magnesium zinc ferrite.

where N is greater than 4. A 1-in-256 selector using seventeen 16-leg structures is being built at present.

The control and drive windings are wound on the structure in a manner independent of the output functions to be generated, and the currents in these windings have lower bounds only. The power required in the control or logic path is maintained small relative to the drive power, as a consequence of the balanced feature of the magnetic circuit. The resultant compact circuit has a large logical capacity.

The practical size of a single sheet will be dictated by leakage flux and fabrication difficulties. Also, as the size increases, both control and drive power will increase.

Acknowledgment

The authors are indebted to E. M. MacKinnon for performing the major portion of the experimental work and for suggesting the balanced holding scheme shown in Fig. 1(b), to R. A. Chegwidden for supplying the ferrite used in the structures, and to L. K. Degen and J. Muller for supplying the tools needed for cutting the structures.

On the State Assignment Problem for Sequential Machines. I*

J. HARTMANIS†

Summary—**In this paper, the problem of determining economical state assignments for finite-state sequential machines is studied. The fundamental idea in this study is to find methods for selection of these assignments in which each binary variable describing the new state depends on as few variables of the old state as possible. In general, these variable assignments in which the dependence is reduced yield more economical implementation for the sequential machine than the assignments in which the dependence is not reduced. The main tool used in this study is the partition with the substitution property on the set of states of a sequential machine. It is shown that for a sequential machine the existence of assignments with reduced dependence is very closely connected with the existence of partitions with the substitution property on the set of states of the machine. It is shown how to determine these partitions for a given sequential machine and how they can be used to obtain assignments with reduced dependence.**

Introduction

In the design of finite-state sequential machines, an important step is the assignment of binary variable states to represent the internal states of the machine. In general, different assignments result in different logical relationships and the complexity of the resulting implementation can vary extensively from assignment to assignment. Since the number of the possible assignments grows extremely rapidly with the number of the internal states of the machine,[1] it is impossible, even with the aid of high-speed digital computers, to try all possible assignments in order to find an economical one for machines with more than twenty states.

In this paper, we study the problem of determining economical state assignments for finite-state sequential machines. The fundamental idea in this study is to find methods for the selection of these assignments in which each binary variable describing the new state depends on as few variables of the old state as possible. Often, these variable assignments in which the dependence is reduced yield more economical implementation for the sequential machine than the assignments in which the dependence is not reduced. The main tool used in this study is the partition with the substitution property on the set of states of a sequential machine. It is shown that for a sequential machine M, the existence of a partition with the substitution property is very closely connected with the existence of assignments of binary state variables in which a subset of these variables can be computed independently from the remaining variables. It is shown how to determine these partitions for a given sequential machine and how they can be used to obtain assignments with reduced dependence. It is also shown how to minimize the number of states of a sequential machine by means of the partitions with the substitution property. This allows us with the same computation to minimize the number of states of a sequential machine and to obtain an assignment with reduced dependence if it exists.

Assignments with Self-Dependent Subsets

Let M be a finite-state sequential machine with the set of internal states $\{S_1, S_2, \cdots, S_n\}$, the finite set of inputs $\{I_1, I_2, \cdots, I_m\}$, and the set of outputs $\{O_1, O_2, \cdots, O_n\}$. The outputs are not necessarily all distinct. We can describe the behavior of M by a flow table as given in Fig. 1. The left-hand column indicates the present state, the top row the present input, and the entries in the table give the corresponding next state. The last row labeled z indicates the output for each state of the machine. (We have assumed that the machine is completely specified; that is, "don't care" conditions do not exist.)

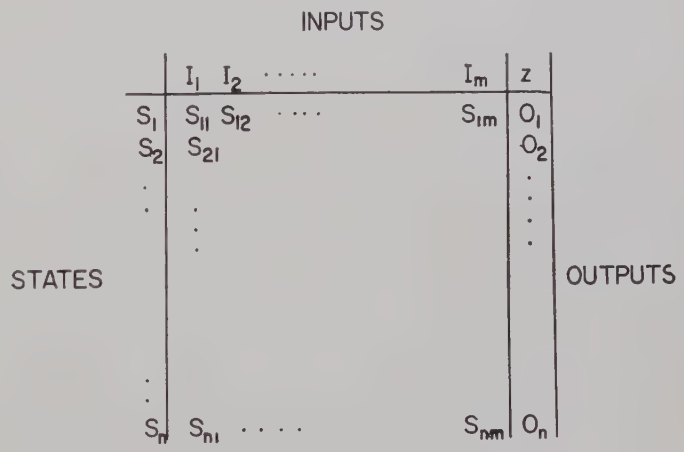


Fig. 1—Flow table for a sequential machine.

To realize a sequential machine, we have to assign some variables to represent the internal state of the machine. We shall study the assignment of binary variables, though most of the ideas developed in this paper can be extended to other assignments. If the sequential machine has n distinct states, and if we wish to make all the assignments of the same length, then we need s

* Received by the PGEC, November 5, 1960; revised manuscript received, December 28, 1960.

† G. E. Res. Lab., Schenectady, N. Y.

[1] E. J. McCluskey and S. H. Unger, "A note on the number of internal variable assignments for sequential switching circuits," IRE Trans. on Electronic Computers, vol. EC-8, pp. 439–440; December, 1959.

binary digits, where s is the smallest integer such that $s \geqq \log_2 n$; we shall write $s = [\log_2 n]$. Our problem is to pick from the many possible assignments one which is easy to realize.[2]

We shall approach this problem by searching for those assignments in which one or more proper subsets of the s binary variables can be computed without the knowledge of the remaining binary variables. In general, if $y_1, y_2, \cdots, y_s$ are the binary variables describing the internal state of the machine, then the ith variable y_i of the next state will depend on the input and all the variables describing the old state, $y_i(t+1) = f_i[y_1(t), y_2(t), \cdots, y_s(t), I(t)]$. [To simplify the expressions, we shall omit the time variable and write $y_i = f_i(y_1, y_2, \cdots, y_s, I)$, since there is little possibility for ambiguity.] As stated above, we are trying to determine those assignments of the variables for which a proper subset of these variables, say, $y_1, y_2, \cdots, y_k$, $1 \leqq k < s$, can be computed without the knowledge of the remaining state variables, that is, $y_i = f_i(y_1, y_2, \cdots, y_k, I)$, $i = 1, 2, \cdots, k$. We shall refer to such assignments as assignments with self-dependent subsets.

Very often, for these machines which have assignments in which some proper subsets of the variables can be computed independently, the state assignments utilizing this possibility of reducing the dependence result in simpler implementation than for those assignments which do not utilize this property of the machine.

It may happen that a sequential machine does not have any assignment with self-dependent subsets of variables, but that there still exist possibilities of decreasing the dependence. For example, the sequential machine may have an assignment $y_1, y_2, \cdots y_k, \cdots y_r, \cdots y_s$, such that $y_1, y_2, \cdots y_k$ depend only on the input and $y_r, y_{r+1}, \cdots y_s$. Since the existence of such assignments cannot be determined by means of partitions with the substitution property, these assignments will be discussed in a sequel to this paper, "On the State Assignment Problem for Sequential Machines. II," in which the required methods will be developed.

Two Assignments for Machine A

To illustrate the basic ideas, we shall consider two different assignments for the sequential machine given in Fig. 2. Since the machine has six internal states, we need three binary variables, y_1, y_2, y_3, to designate these states. We shall consider two binary variable assignments α and β given in Fig. 3.

If we substitute these binary variable assignments in the flow table given in Fig. 2, we obtain the relations shown in Fig. 4. For the assignment α the binary variables of the new state can be computed from the variables of the old state and the input x as follows:

$$y_1 = \bar{y}_1\bar{y}_2y_3\bar{x} + y_2\bar{y}_3\bar{x} + y_2y_3x$$

$$y_2 = \bar{y}_2x + \bar{y}_1\bar{y}_2\bar{y}_3 + y_1y_3$$

$$y_3 = y_1x + y_2\bar{y}_3x + y_2y_3\bar{x} + \bar{y}_1\bar{y}_2\bar{x}.$$

It is seen that these equations are fairly complicated and that every variable of the new state depends on all the variables of the old state and the input.

For the assignment β the relations between the new and old state variables are given by the following equations:

$$y_1 = y_1x + \bar{y}_1\bar{x}$$

$$y_2 = y_3\bar{x}$$

$$y_3 = \bar{y}_2\bar{y}_3.$$

It can be seen that y_1 does not depend on y_2 or y_3 and that y_2, y_3 does not depend on y_1. Furthermore, these equations are far simpler than the equations relating the new and old state variables for the assignment α.

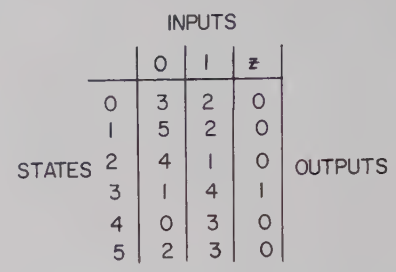


Fig. 2—Sequential machine A.

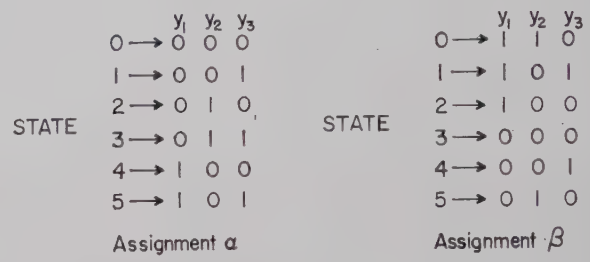


Fig. 3—Assignments α and β for machine A.

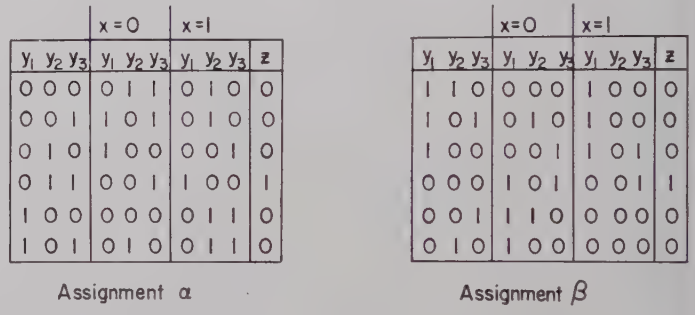


Fig. 4—Flow tables for the assignments α and β.

[2] *Ibid.*, the authors have pointed out that this is an exceedingly difficult problem. It should be mentioned that different measures can be applied to compare the quality of assignments and that these will depend on the logical elements used and the restrictions imposed on the configurations of these elements. In this paper we have restricted ourselves to the use of two layer AND and OR gate realizations in which we can easily count the number of diodes. On the other hand, since we minimize the number of dependencies between the variables of the assignment, we feel that these methods will yield economical realizations for a wider class of components and configurations.

The resulting switching circuit for the assignment α is about three times more complicated than the circuit resulting from the assignment β.

An implementation of this machine utilizing the assignment β is shown in Fig. 5. It should be observed that in this case the realization of A consists of two independent machines A_1 and A_2 operating in parallel.[3] We shall return to this example in the following section.

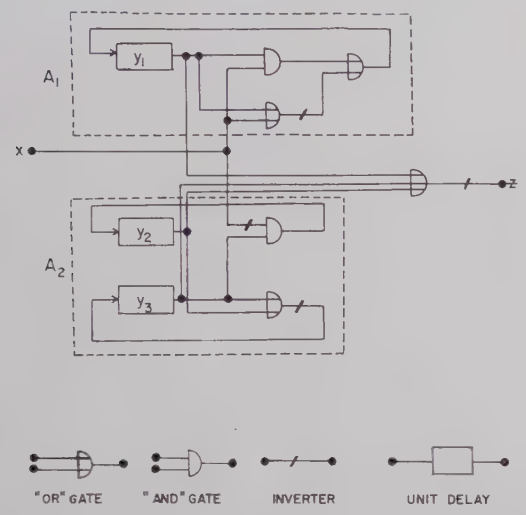


Fig. 5—Realization of machine A using assignment β.

Partitions and Assignments with Self-Dependent Subsets

We shall now study the properties of the assignment β in order to be able to select such assignments systematically. The main problem is to find characteristics of the machine which show that there exist assignments in which some set of the variables describing the state can be computed without the knowledge of the remaining ones. For the example of Fig. 2 and the assignment β given in Fig. 4, there exist two such sets, the set consisting of y_1 and the set consisting of y_2 and y_3. When we consider only the variable y_1, then we cannot distinguish between the states 0, 1 and 2, or 3, 4, and 5. On the other hand, since y_1 can be computed if we know the old y_1 and the input x, after any number of inputs the resulting y_1 will tell us in which one of the two sets of states the state of A will be contained. Thus, the ignorance about the actual state of the machine does not increase. Similarly, if we consider only the two last variables y_2 and y_3, we cannot distinguish between the states 0 and 5, or 1 and 4, or 2 and 3. Again, if we know y_2 and y_3 and know the following input sequence, we can compute the resulting y_2 and y_3. That is, if we know in which one of the three possible sets the initial state is contained, and if we know the input sequence, we can determine the set which will contain the resulting state.

[3] J. Hartmanis, "Symbolic analysis of a decomposition of information processing machines," *Information and Control*, vol. 3, pp. 154–178; June, 1960.

In both cases, we can partition the set of states into disjoint subsets in such a way that, if we know in which subset the starting state is contained, we are able to tell in which of these sets the state will be contained after any finite sequence of inputs. To formalize these ideas, we shall make some definitions.

Definition 1

A partition π on a set S is a collection of disjoint subsets of S such that their set union is S.

The subsets of the partition π on S will be called the blocks of the partition and we shall describe π by listing these blocks. Thus, the two partitions π_1 and π_2 in our example [determined by (y_1) and (y_2, y_3), respectively] are represented by $\pi_1 = \{\overline{0, 1, 2}; \overline{3, 4, 5}\}$ and $\pi_2 = \{\overline{0, 5}; \overline{1, 4}; \overline{2, 3}\}$. We shall say that the partition π identifies two elements if and only if these elements are contained in the same block. We shall refer to the partition $\pi = 0$, in which each block consists of a single element, and to the partition $\pi = I$, in which all elements are contained in one block as the trivial partitions.

Definition 2

A partition π on the set of states of a sequential machine M is said to have the substitution property (S.P.) with respect to M, if for any two states S_i and S_j belonging to the same block of π and any input I, the states IS_i and IS_j are again contained in a common block of π. (IS_i is the state the machine goes into from S_i when the input I is applied.)

Note that the partitions $\pi_1 = \{\overline{0, 1, 2}; \overline{3, 4, 5}\}$ and $\pi_2 = \{\overline{0, 5}; \overline{1, 4}; \overline{2, 3}\}$ have the substitution property with respect to the machine A given in Fig. 2. The following theorem states the general case establishing a relation between assignments with a self-dependent subset of variables and partitions with the substitution property.

Theorem 1: Let M be a sequential machine with n states to each of which are assigned one of the distinct sequences of s binary variables, $y_1, y_2, \cdots y_s$. If the first k variables, $1 \leqq k < s$, designating the next state, can be computed from the input and the first k variables of the old state, then there exists a partition π with the substitution property for M which identifies all the states which agree in the first k variables.

To prove this theorem, we first observe that we obtain a partition π if we form subsets of the states of M by placing in the same subset all the states which agree in the first k variables. That is, every state is contained in one and only one of these subsets or blocks. To verify that this partition π has the substitution property, we recall that any two states S_i and S_j which are contained in the same block have the same first k digits in their description. Since the first k variables of the next states which follow S_i and S_j, respectively, are determined by the input and the first k variables of the states S_i and S_j, we see that these k variables will be identical for the next states of S_i and S_j, if the same input is used.

Therefore, these states will again be contained in a common block of π, which shows that π has the substitution property with respect to M. (Note that this theorem applies to the binary assignments, but it is not restricted to such assignments.)

The previous theorem shows that if there exists an assignment with a proper self-dependent subset, then there exists a corresponding partition with S.P. If we use binary variables to designate the internal states of the machine, then there may exist partitions with S.P. which cannot be utilized to obtain assignments with self-dependent subsets for the minimum number of variables. Let us now see which partitions with S.P. lead to such assignments and how such assignments are made.

Consider a sequential machine M with n internal states and let π be a partition with the substitution property on the set of states of M. (We assume that the number of internal states of M is minimized.) Denote the number of distinct blocks of π by $\#(\pi)$ and the number of distinct states in the largest block of π by $m(\pi)$. Since π has the substitution property, we know that we can compute the block in which the next state will be contained if we know the input and the block in which the present state is contained. Thus, we can assign the first $k = \lceil \log_2 \#(\pi) \rceil$ digits to distinguish between the blocks of π and compute these without the knowledge of the remaining digits. (See assignment β for machine A.) The remaining digits have to be used to distinguish between the states contained in a block. Since the largest block contains $m(\pi)$ states, we need $\lceil \log_2 m(\pi) \rceil$ digits to distinguish the states within a block. Thus to utilize the partition π in assigning the state variables, we need $\lceil \log_2 \#(\pi) \rceil + \lceil \log_2 m(\pi) \rceil$ binary digits. If this is done, then the first k digits can be computed without the knowledge of the remaining digits. This can be summarized in the following theorem.

Theorem 2: Let M be a sequential machine with n internal states. Then the existence of a nontrivial partition π with S.P. implies that there exists an assignment of s, $s = \lceil \log_2 n \rceil$, binary digits to the states of M such that the first k digits, $1 \leqq k < s$, of the next state can be computed without the knowledge of the last $s-k$ digits of the old state, if

$$\lceil \log_2 \#(\pi) \rceil + \lceil \log_2 m(\pi) \rceil = s.$$

To illustrate this, consider the sequential machine B given in Fig. 6. This machine has seven states so that we need three binary digits to represent these states. Note that the two partitions $\pi_1 = \{\overline{0, 1}; \overline{2, 3}; \overline{4, 5}; \overline{6}\}$ and $\pi_2 = \{\overline{0, 1}; \overline{2, 3}; \overline{4, 5, 6}\}$ have the substitution property for the machine B. The partition π_1 has four blocks and the largest number of elements in a block is two. Thus,

$$\lceil \log_2 \#(\pi_1) \rceil + \lceil \log_2 m(\pi_1) \rceil = 3,$$

and we can utilize π_1 to obtain a three-digit assignment in which the two first variables distinguish between the blocks of π_1 and thus can be computed independently of the third one. The third digit will distinguish between the states in the blocks of π_1, and it may depend on the input and all three state variables. On the other hand, π_2 has three blocks and the largest number elements in the block is also three. Thus,

$$\lceil \log_2 \#(\pi_2) \rceil + \lceil \log_2 m(\pi_2) \rceil = 4,$$

and therefore, we would have to have a four-digit assignment to exploit π_2. If we do this then the first two digits can be computed independently of the last two, but the last two will, in general, depend on all four digits and the input.

STATES	INPUTS x=0	INPUTS x=1	OUTPUTS z
0	5	2	1
1	4	3	0
2	1	4	1
3	0	5	0
4	2	6	0
5	3	6	0
6	2	4	1

Fig. 6—Machine B.

The two previous theorems show the connection between the assignments with a proper self-dependent subset and the partitions with S.P. on the set of states of a sequential machine. We shall now show how to obtain the partitions with the substitution property for a given machine.

To determine the partitions with S.P. for a given machine M, we can start by identifying any two distinct states, say S_1 and S_2. Then we have to identify the pairs of states S_{1k} and S_{2k} in which S_1 and S_2 go if we apply the kth input, $k = 1, 2, \cdots, m$. To this set of pairs, we add those pairs which can be identified by the transitive law; that is, if S_i and S_j are identified and so are S_j and S_k, we have to identify S_i and S_k. We now repeat the process by looking up the new identifications induced by the new pairs. If after R steps, the $R+1$ step does not yield any new identifications, we have constructed a partition π on S which has the S.P. on M. Note that if there does not exist a nontrivial partition which has the S.P., then this process will stop after identifying all states of M. For a machine M with n states, we have to try the $n(n-1)/2$ distinct pairs of states to determine that there does not exist a nontrivial partition with S.P. for M. Furthermore, it is shown later in this paper that we can generate all partitions with S.P. for M by combining the partitions obtained, by the above-described method, from the identifications of pairs of states.

To illustrate the computation of partitions with S.P., let us consider the machine A given in Fig. 2. If we identify states 0 and 1, then the input "0" leads to states 3 and 5, respectively, and we have to identify these states. Note that the input "1" does not yield any new identifications because the states 0 and 1 both go into

the state 2 under this input. Since the pairs of states 0, 1 and 3, 5 are disjoint, we do not have to add any new pairs because of the transitive law. Repeating the process for the states 3 and 5, we obtain the pairs 1, 2 and 4, 3. This time by the transitive law, we obtain that the states 0, 1, 2 are identified and 3, 4, 5 are also identified. It is seen that no further identifications are needed.

Thus, $\pi_1 = \{\overline{0, 1, 2}; \overline{3, 4, 5}\}$ has the substitution property. Similarly, we can obtain $\pi_2 = \{\overline{0, 5}; \overline{1, 4}; \overline{2, 3}\}$. On the other hand, it can be seen that if we identify any pair of states not identified by π_1 or π_2, for example, the states 1 and 5, we are forced to identify all the states of A. Thus, this machine has only two nontrivial partitions with the substitution property on its set of states.

Algebraic Properties of Partitions

So far, it has been shown how to obtain, for a given machine, partitions with the substitution property and how to utilize them to assign binary codes to the state variables in order to obtain an assignment with a self-dependent subset of variables. We shall now show how we can generate new partitions with S.P. by combining other partitions which have S.P. The knowledge of these methods may decrease the amount of work which has to be done to determine partitions which lead to economical state assignments. Furthermore, it will be seen from the examples at the end of this section and the following section, that these ideas are also essential if we want to find assignments which have several self-dependent subsets of variables. Since each self-dependent subset of variables corresponds to a partition with S.P., the detection of assignments with more than one such subset involves consideration of several partitions with S.P. and their mutual relations. (For a more detailed discussion of the algebraic properties of partitions, see Hartmanis[3] and Birkhoff.[4])

First, we shall define an ordering (partial ordering) for partitions on a set S and show how to combine partitions. After that, we shall discuss the preservation of the substitution property by the combining operations.

We shall say that a partition π_1 on S is smaller than or equal to π_2 on S, $\pi_1 \leqq \pi_2$, if and only if each block of π_1 is contained in a block of π_2. Thus, the smallest partition on S is the partition in which each block consists of a single element. The largest partition on S has only one block and this block is S. We referred to these two partitions as trivial and denoted them by O and I, respectively.

Under the above-defined ordering of the partitions on a set S there are pairs of partitions which cannot be compared; that is, for some π_1 and π_2 we have that $\pi_1 \not\leqq \pi_2$ and $\pi_2 \not\leqq \pi_1$. In such cases, it is useful to be able to construct the least upper bound and the greatest lower bound for these partitions. The least upper bound (l.u.b.) of any two partitions π_1 and π_2, is a partition π_3 such that $\pi_1 \leqq \pi_3$, $\pi_2 \leqq \pi_3$ and if $\pi_1 \leqq \pi_4$, $\pi_2 \leqq \pi_4$, then $\pi_3 \leqq \pi_4$. Dually the g.l.b. is defined for π_1 and π_2. We shall denote the l.u.b. of π_1 and π_2 by $\pi_1 + \pi_2$ and the g.l.b. by $\pi_1 \cdot \pi_2$. The following shows how to construct for π_1 and π_2 the partitions $\pi_1 \cdot \pi_2$ and $\pi_1 + \pi_2$. Given π_1 and π_2, then the blocks of $\pi_1 \cdot \pi_2$ are obtained by intersecting the blocks of π_1 and π_2. Thus, S_i and S_j of S are contained in the same block of $\pi_1 \cdot \pi_2$ if and only if S_i and S_j are contained in the same block in π_1 and π_2. To construct the blocks of $\pi_1 + \pi_2$, we shall introduce some notation.

If a family of sets is given, then we say that two sets A and B from this family are connected if their intersection is nonvoid, $A \wedge B \neq \phi$. Two sets of this family, say A and C, are said to be chain connected in this family of sets if there exists a sequence of sets $A = A_1$, A_2, A_3, $\cdots$, $A_n = C$ such that A_i is connected to A_{i+1}, $i = 1, 2, \cdots, n-1$.

If two partitions π_1 and π_2 are given and A is a block of π_1, then in the partition $\pi_1 + \pi_2$ the block which contains A is the set union of all blocks of π_1 and π_2 which are chain connected to A in the family of subsets consisting of the blocks of π_1 and π_2. Again it is not hard to check that the collection of all such blocks form a partition and that this partition is the l.u.b. of π_1 and π_2. To illustrate the construction of $\pi_1 \cdot \pi_2$ and $\pi_1 + \pi_2$ let us consider a set S with nine elements, say $S = \{a, b, c, \cdots, i\}$, and let

$$\pi_1 = \{\overline{a, b}; \overline{c, d}; \overline{e, f}; \overline{g, h, i}\}$$

and

$$\pi_2 = \{\overline{a, f}; \overline{b, c}; \overline{d, e}; \overline{g, h}; \overline{i}\}.$$

Then

$$\pi_1 \cdot \pi_2 = \{\overline{a}; \overline{b}; \overline{c}; \overline{d}; \overline{e}; \overline{f}; \overline{g, h}; \overline{i}\}$$

and

$$\pi_1 + \pi_2 = \{\overline{a, b, c, d, e, f}; \overline{g, h, i}\}.$$

The importance of the above-discussed operations for the study of sequential machines is contained in the fact that these operations preserve the substitution property for a sequential machine as stated in the following theorem.

Theorem 3: If π_1 and π_2 are two partitions with the substitution property on the set of states of a sequential machine M, then

$$\pi_1 \cdot \pi_2$$

and

$$\pi_1 + \pi_2$$

have the substitution property for M.

The proof of this theorem is very similar to the proof given in Hartmanis[3] or Birkoff[4] for binary relations and will not be given here. This theorem shows that the collection of all partitions with S.P. on the set of states of a sequential machine forms a lattice under the above-defined ordering (see Birkhoff[4]). We shall return to this concept in the discussion of the next example.

[4] G. Birkhoff, "Lattice theory," *Am. Math. Soc.*, vol. 25; 1948.

Note that this theorem allows us to generate new partitions with S.P. from known ones without using the flow table. Furthermore, if we have all the partitions which are obtained by identifying a pair of states of a sequential machine and seeing what partition this identification induces, we can generate all other partitions with S.P. for M from this set of partitions. To do this, we just have to combine the partitions by the $+$ operation.

Corollary 1: Let $\pi(S_i, S_j)$ denote the smallest partition with S.P. for M which identifies the states S_i and S_j. Then any partition π on the set of states of M with S.P. is given by a sum over a subset of the partitions $\pi(S_i, S_j)$.

To see this, note that if π has S.P., then partition sum

$$\sum_{i,j} \pi(S_i, S_j) = \pi,$$

where the summation is over the i, j such that $\pi(S_i, S_j) \leqq \pi$.

To show an example for the application of the above-developed operations to the problem of determining an assignment with several self-dependent subsets of variables, we shall consider the machine C, given in Fig. 7.

STATES	INPUTS x=0	INPUTS x=1	OUTPUTS z
0	3	6	0
1	2	7	1
2	4	5	0
3	5	5	0
4	7	2	0
5	6	3	0
6	0	1	0
7	1	1	0

Fig. 7—Machine C.

This machine has eight states, and thus we need three binary variables to designate the states of the machine. To find assignments with self-dependent subsets we shall construct partitions with S.P.

When we identify the states 0 and 1, we obtain the partition

$$\pi_1 = \{\overline{0,1}; \overline{2,3}; \overline{4,5}; \overline{6,7}\}.$$

According to Theorem 2, we can utilize this partition to obtain a three-variable assignment with a self-dependent subset of variables. To do this we could assign the first two variables y_1 and y_2 to distinguish between the four different blocks of π_2 and let the third variable distinguish between the two elements in each block. In this case y_1 and y_2 will depend only on themselves and the input, but y_3 can depend on y_1, y_2, y_3 and the input. We shall now construct additional partitions with S.P. and show how the dependence can be reduced further.

The identification of the states 0 and 2 yields the trivial partition $\pi = I$, and so do the identification of the states 0 and 3. The identification of 0 and 4 yields the partition

$$\pi_2 = \{\overline{0,4}; \overline{1,5}; \overline{2,6}; \overline{3,7}\}.$$

Again this partition alone could be used to obtain an assignment with a self-dependent subset of variables. On the other hand, it is possible to use both partitions to obtain an assignment with more than one self-dependent subset. If we use four binary variables we can utilize both partitions by assigning y_1, y_2 to distinguish between the blocks of π_1 and y_3, y_4 to distinguish between the blocks of π_2. In this case y_1, y_2 and y_3, y_4 form two disjoint self-dependent subsets. (Furthermore, it is seen that if we know in which block of π_1 and π_2 the state of the machine is contained we know the exact state, since $\pi_1 \cdot \pi_2 = 0$.) Actually, we can replace the four-variable assignment with a three-variable assignment because we can let the two assignments distinguishing between the blocks of π_1 and π_2 share a variable. To see this we observe that

$$\pi_1 + \pi_2 = \{\overline{0,1,4,5}; \overline{2,3,6,7}\}.$$

Because of Theorem 3 we know that $\pi_1 + \pi_2$ have S.P. Thus we can utilize $\pi_1 + \pi_2$ to assign one variable, say y_1, to distinguish between the blocks of $\pi_1 + \pi_2$ (as shown in Fig. 8).

We now observe that π_1 and π_2 each splits the blocks of $\pi_1 + \pi_2$ in two parts. Thus, we can assign the second digit, y_2, to distinguish with y_1 and y_2 between the blocks of π_1, and similarly a third digit to distinguish with y_1 and y_3 between the blocks of π_2. One such assignment is given in Fig. 8.

STATES	y_1	y_2	y_3
0 →	0	0	1
1 →	0	0	0
2 →	1	0	1
3 →	1	0	0
4 →	0	1	1
5 →	0	1	0
6 →	1	1	1
7 →	1	1	0

Fig. 8—Assignment for machine C.

The relations between the old and new variables are given for this assignment by the following:

$$y_1 = \bar{y}_1$$
$$y_2 = y_1\bar{y}_2 + \bar{y}_1 y_2 \bar{x} + \bar{y}_2 x$$
$$y_3 = \bar{y}_1 y_3 x + \bar{y}_1 \bar{y}_3 \bar{x} + y_1 y_3 \bar{x}.$$

The resulting realization is shown in Fig. 9. (In these realizations no attempt has been made to minimize the number of complements.) In this example we obtained

an assignment with three overlapping self-dependent subsets of variables. So far we do not have any systematic method of picking out the assignment with the largest number of self-dependent subsets if they are allowed to overlap. On the other hand, the problem for the case when the subsets are disjoint is easier and is discussed in the next section.

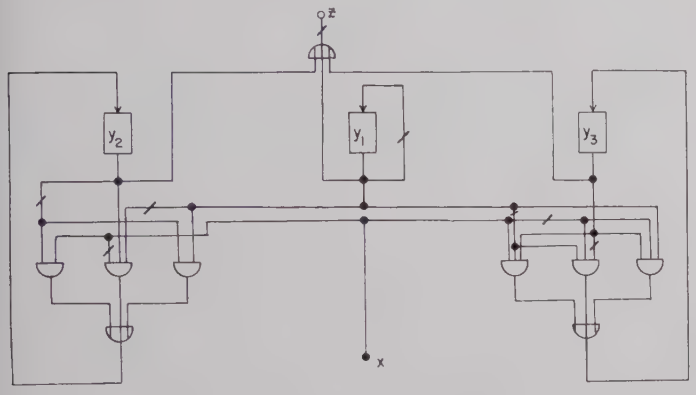


Fig. 9—Realization of machine *C*.

Decomposition of Sequential Machines

We shall now derive conditions on two partitions with S.P. under which there exist assignments such that the s binary digits needed to describe the n internal states of the machine M can be split into two parts such that each part can be computed independently. Since such an assignment exists for the machine A, we shall use this machine as an example.

Observe that for the machine A the two partitions

$$\pi_1 = \{\overline{0,1,2};\overline{3,4,5}\} \quad \text{and} \quad \pi_2 = \{\overline{0,5};\overline{1,4};\overline{2,3}\}$$

are such that every block of π_1 has exactly one element in common with every block from π_2. Thus, if we know in which block of π_1 and in which block of π_2 the state is contained, we know exactly which state it is. Note that this fact was exploited in assigning the codes in the case β. The first digit was used to determine in which of the two blocks of π_1 the state is contained, and the last two digits are used to determine in which one of the three possible blocks of π_2 the state of the machine is contained. Since these partitions have the substitution property we can compute the first digit and the last two digits independently. This is reflected in the realization of A by two independent machines A_1 and A_2 which operate in parallel (Fig. 5). To be able to achieve such a decomposition in two machines, we must have two nontrivial partitions π_1 and π_2 with S.P. such that no two distinct states of M are identified by both π_1 and π_2. In other words, a block from π_1 and π_2 can have at most one element in common ($\pi_1 \cdot \pi_2 = 0$). If this is the case, then we can assign k_1+k_2 binary digits so that the first k_1 digits, $k_1 = \lceil \log_2 \#(\pi_1) \rceil$ distinguish between the blocks of π_1 and that the last k_2 digits, $k_2 = \lceil \log_2 \#(\pi_2) \rceil$ distinguish between the blocks of π_2. If we know the input and the first k_1 digits, then, because of the substitution property, we can compute the first k_1 digits describing the next state. Similarly, the last k_2 digits can be computed independently of the first k_1 digits. These results are summarized in the following theorem.

Theorem 4: A sequential machine M with n states has a binary variable assignment of length $s = \lceil \log_2 n \rceil$, which can be split into two parts, such that the first k variables $1 \leqq k < s$, and the last $s-k$ variables can be computed independently, if and only if there exist two nontrivial partitions π_1 and π_2 with S.P. for M satisfying the two conditions:

1) Any block of π_1 has at most one element in common with any block of π_2 ($\pi_1 \cdot \pi_2 = 0$).
2) $\lceil \log_2 \#(\pi_1) \rceil + \lceil \log_2 \#(\pi_2) \rceil = s$.

It was pointed out previously that the machine A satisfied both conditions of this theorem and thus could be realized by two machines operating in parallel as shown in Fig. 5.

It should be observed that the definition of decomposition given in Hartmanis[3] does not admit the machine A as decomposable since the two machines A_1 and A_2 have the same input. It seems that this is too strong a requirement, and we shall refer to a machine as decomposable into two parts for a binary assignment if the conditions of Theorem 4 are satisfied.

If condition 1) of Theorem 4 is satisfied, but condition 2) is not, then we can decompose the machine in two independent machines only by introducing additional variables in the state assignment. Machine D described in Fig. 10 illustrates this case.

INPUTS

STATES	$x_1 x_2$ = 0 0	0 1	1 0	1 1	z
0	1	2	3	0	0
1	0	2	3	0	1
2	1	0	3	0	1
3	1	2	0	0	1

OUTPUTS

Fig. 10—Machine *D*.

This machine has six nontrivial partitions with S.P.:

$$\pi_1 = \{\overline{0,1};\overline{2};\overline{3}\}, \quad \pi_2 = \{\overline{0,2};\overline{1};\overline{3}\}, \quad \pi_3 = \{\overline{0,3};\overline{1};\overline{2}\}.$$

$$\pi_4 = \{\overline{0,1,2};\overline{3}\}, \quad \pi_5 = \{\overline{0,2,3};\overline{1}\} \text{ and } \pi_6 = \{\overline{0,1,3};\overline{2}\}.$$

It can be seen that no two of these partitions satisfy condition 1) and 2) of Theorem 4, though several pairs satisfy the first condition, for example, π_1 and π_5.

Since

$$\log_2 \lceil \#(\pi_1) \rceil + \log_2 \lceil \#(\pi_5) \rceil = 3,$$

we need one additional digit to realize machine C by two machines operating in parallel. Before proceeding with the design, we should observe that the machine

resulting from π_1 can be split further into two independent machines. This could have been seen right away by considering the three partitions π_4, π_5, and π_6 and observing that

$$\pi_4 \cdot \pi_5 \cdot \pi_6 = 0.$$

We can now assign y_1 to distinguish between the blocks of π_4; y_2 to distinguish between the blocks of π_6 and y_3 for π_5. One such assignment is given in Fig. 11.

The resulting relations between the new and old state variables are given below.

$$y_1 = \bar{y}_1 x_1 \bar{x}_2$$
$$y_2 = \bar{y}_2 \bar{x}_1 x_2$$
$$y_3 = \bar{y}_3 \bar{x}_1 \bar{x}_2.$$

The resulting realization is shown in Fig. 12.

	y_1	y_2	y_3
0 →	0	0	0
1 →	0	0	1
STATES 2 →	0	1	0
3 →	1	0	0

Fig. 11—A three-variable assignment for machine *D*.

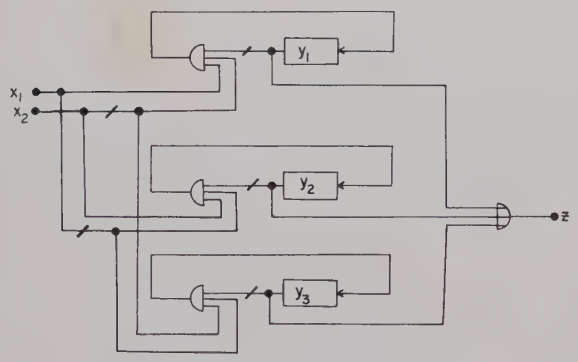


Fig. 12—Realization of machine *D*.

In this case, we have three machines operating in parallel. Any assignment utilizing two variables requires considerably more logic than this three variable assignment.

State Reduction

Usually in the design of sequential machines, once the flow table is obtained the superfluous states are eliminated to simplify the design.[5-7] In this section we shall show how the state reduction can be combined with the search for assignments with self-dependent subsets. (We have assumed that the machine is completely specified; that is, there do not exist "don't care" conditions.)

Definition 3

A partition on the set of states of a sequential machine M will be said to be output consistent, if for every block of π all the states contained in the block have the same output.

The output-consistent partitions for a sequential machine preserve this property under the two laws of partition combination defined previously. This is stated in the following Corollary and is a parallel of Theorem 3.

Corollary 2: If π_1 and π_2 are two output-consistent partitions on the set of states of a sequential machine M, then $\pi_1 \cdot \pi_2$ and $\pi_1 + \pi_2$ are also output consistent.

To prove this we recall that any block of $\pi_1 \cdot \pi_2$ is obtained by intersecting a block A of π_1 and a block B of π_2. Since all the states in the block A have the same output, any subset of A will also contain states with the same output. Thus, the block $A \wedge B$ is output consistent, and since this holds for any pair of blocks of π_1 and π_2, we conclude that $\pi_1 \cdot \pi_2$ is output consistent. Similarly, to verify the output consistence for $\pi_1 + \pi_2$, we recall that a block of $\pi_1 + \pi_2$ is the set union of chain connected blocks of π_1 and π_2. But if two blocks, A from π_1 and B from π_2, have some state in common, then all the states in A must have the same output as all the states in B. Thus, any block formed from connected blocks of π_1 and π_2 will be output consistent since π_1 and π_2 are output consistent. From which we conclude that $\pi_1 + \pi_2$ is output consistent.

If we want to identify any states of a sequential machine without changing the input-output relation of this machine, we can only identify states with the same output. Thus, the sets of states which will be identified to reduce the number of states of a machine will form an output-consistent partition. Furthermore, this partition has to have the substitution property, since otherwise the resulting machine will not have unique next states. From these observations, we see that, for a given machine M, we can reduce the number of states if there exists a nontrivial partition on the set of states of M with S.P. which is output consistent. The states of the reduced machine are just the blocks of states of M under this partition.

Furthermore, since this partition preserves S.P. and output consistency under the + operation, we have a straightforward way to obtain the machine with the minimal number of states. To do this we consider all the partitions with S.P. which result from the identification of a pair of states $\pi(S_i, S_j)$. From this set we select those partitions which are output consistent. If there are not any which are output consistent, the machine cannot be reduced. If there are partitions $\pi(S_i, S_j)$ which are output consistent, the sum partition of this set will be the partition giving the maximal reduction possible.

[5] D. A. Huffman, "The synthesis of sequential switching circuits," *J. Franklin Inst.*, vol. 257, pp. 151–190, 275–303; March and April, 1954.

[6] E. F. Moore, "Gedanken-experiments on sequential machines," in "Automata Studies," C. E. Shannen and J. McCarthy, Eds., Princeton University Press, Princeton, N. J.; 1956.

[7] G. H. Mealy, "A method for synthesis of sequential circuits," *Bell Sys. Tech. J.*, vol. 34, pp. 1045–1079; September, 1955.

To illustrate this, we shall consider machine C described in Fig. 7. For this machine, we know all the partitions with S.P. It is seen that none of these partitions is output consistent; therefore, we see that this machine cannot be reduced. On the other hand, if we change the output for the state 0 to a 1, we obtain a nontrivial output consistent partition with S.P.,

$$\pi = \{\overline{0, 1}; \overline{2, 3}; \overline{4, 5}; \overline{6, 7}\}$$

This is the only nontrivial output consistent partition with S.P., and we see that four different pairs of states can be identified, and thus the resulting four-state machine will have the same input-output relations as the original eight-state machine.

Conclusion

In this paper, we discussed a method of matching the binary variable assignments for the internal states of a sequential machine to the structure of this machine in order to obtain economical realizations. The fundamental idea in this study was to decrease the number of old state variables on which the variables of the next state depend. We did this by finding subsets of variables which could be computed without the knowledge of the remaining state variables. It was shown how to obtain such assignments and that these assignments in which dependence is reduced usually result in realizations which are more economical than the realizations for the assignments where the dependence is not reduced.

In another paper, "On the State Assignment Problem for Sequential Machines. II," we shall give methods to determine assignments with reduced dependence which cannot be obtained from partitions with the substitution property.

Acknowledgment

The author would like to acknowledge several of his colleagues whose valuable discussions have aided in the writing of this paper. Among these are C. L. Cotes, P. M. Lewis, and R. L. Shuey of the General Electric Research Laboratory. He should also like to thank F. C. Hennie of the Massachusetts Institute of Technology and R. E. Stearns of Princeton University who spent the last summer at the General Electric Research Laboratory.

Bibliography

[1] R. L. Ashenhurst, "The decomposition of switching functions," in "Proceedings of an International Symposium on the Theory of Switching," Harvard University Press, Cambridge, Mass.; 1959.

A Generalization of a Theorem of Quine for Simplifying Truth Functions*

J. T. CHU†

Summary—**A method of Quine for identifying the core prime implicants of a given truth function, without obtaining all its prime implicants, is generalized under the so-called "don't care" conditions. It is shown that our method is equivalent to, and sometimes an improvement of, a result of Roth. When all the prime implicants (under the don't care conditions) of a truth function are given, our method becomes a generalization of a result of Ghazala and is equivalent to another result of Roth. It is also pointed out that our method may be used, in a way similar to using Roth's, for simplifying truth functions.**

Introduction

In a recent paper, Quine[1] obtained a method for identifying the core prime implicants of a given truth function. An advantage of his method is that it does not require the derivation of *all* the prime implicants of the given function, which saves time. In this paper, we shall generalize Quine's method under the so-called "don't care" conditions. It will be shown that our method is equivalent to, and sometimes an improvement of, a result obtained by Roth[2] for the same purpose. We will also show that ours is a generalization of a method of Ghazala[3] and is equivalent to another result of Roth[2] for identifying the core prime implicants when all the prime implicants of the truth function are given. In addition, Roth extended his method for identifying the core prime implicants and obtained a method for simplifying truth functions. It will be pointed out that ours can be extended in the same way.

* Received by the PGEC, June 7, 1960; revised manuscript received, January 11, 1961.

† Moore School of Elec. Engrg., University of Pennsylvania, Philadelphia, Pa. This work was performed while the author was employed at RCA, Camden, N. J.

[1] W. V. Quine, "On cores and prime implicants of truth functions," *Am. Math. Monthly*, vol. 66, pp. 755–760; November, 1959.

[2] J. P. Roth, "Algebraic topological methods in synthesis," *Proc. Internatl. Symp. on the Theory of Switching*, Harvard University Press, Cambridge, Mass., pt. 1, pp. 57–73; 1959.

[3] M. J. Ghazala, "Irredundant disjunctive and conjunctive forms of a Boolean function," IBM J. Res. and Dev., vol. 1, pp. 171–176; April, 1957.

To make this paper compact, we shall assume familiarity with Quine's[1] work and use his results freely. On the other hand, concepts and terminologies involving don't care conditions will be precisely defined, even though similar concepts and terminologies have been used by other authors, (*e.g.*, Mott,[4] Nelson,[5] and Roth.[2] We shall begin with a definition and a lemma.

Definition 1: Let F and D be given truth functions of the letters $x_1, x_2, \cdots, x_n$. A truth function G is said to be D-equivalent to F if and only if $D=0 \rightarrow G=F$. ("$\rightarrow$" means "implies.") We write $G=F(D)$. A D-implicant (DI) f of F is a term (Quine's[1] fundamental formula) of the x_i's and $\bar{x}_i$'s, such that $F=1$ whenever $D=0$ and $f=1$. (If $D=0 \rightarrow f=0$, f is also a DI of F.) A D-prime implicant f(DPI) of F is a DI of F which does not properly subsume other DI's g of F. (f properly subsumes g if f subsumes g, but $f \neq g$.)

Some explanations about the notations and definitions perhaps should be given here. If both sides of an equal sign are letters (for example, $G=F$), it means that they are equivalent (or identically equal) to each other. However, if the right-hand side is a number (for example, $f=1$), then it means that f is equal to 1. The identity $f \equiv 1$ is used to denote that f is equivalent to 1. Thus $x_1x_2+x_1\bar{x}_2=x_1$, $x_1x_2=1$ if $x_1=1$ and $x_2=1$, and $x_1\bar{x}_1 \equiv 0$. A few examples will clarify Definition 1. Suppose that $F=x_1\bar{x}_2+x_3$, $D=x_1x_2$, and $G=x_1+x_3$. Then F is D-equivalent to G, for $D=0$ implies that $F=F+D=G$. But F and G are not equivalent to each other. Further, $x_1\bar{x}_2$ is a DI of F, since $x_1\bar{x}_2=1 \rightarrow F=1$. But $x_1\bar{x}_2$ is not a DPI of F, because it properly subsumes x_1 which is also a DI of F. On the otherhand x_1 is a DPI of F.

It should also be remarked that F is a truth function[1] which represents a given switching circuit, while $D=0$ represents the don't care conditions. If $G=F(D)$, then the switching circuits represented by F and G perform the same function for all legitimate combinations of the input signals. Our main interest is to find a normal form[1] $G=F(D)$ which cannot be simplified further. Obviously, the terms in such a G must be DPI's of F. In the following it shall be shown that sums of DPI's do exist which are D-equivalent to F, and further, that there are two kinds of DPI's, one kind which may be replaced by sums of other DPI's, and is consequently not indispensable; another kind, which may not be replaced and must, therefore, be present in all normal forms $G=F(D)$. It has been shown[2] that the simplest normal forms $G=F(D)$ may be obtained by successively identifying the second kind of DPI's. In this paper, several methods for identifying such DPI's are given.

Let H be a sum of the DPI's of F such that $H=F(D)$, and $A(F/D)$, the set of all such H's (for given F and D). The following lemma shows that such H does exist.

Lemma: The set $A(F/D)$ is nonempty.

Proof: It is easy to show that a term f is a DPI of F if, and only if, f is a prime implicant (PI) of $F+D$ (see, *e.g.*, Mott[4]). The sum H of all the PI's of $F+D$ is equivalent to $F+D$[6] and hence $H=F(D)$. Therefore $A(F/D)$ is nonempty.

A Theorem

Definition 2: A DPI of F is a core D-prime implicant ($CDPI$) of F if it is a term of every $H \in A(F/D)$. The D-core of F is the set of all the $CDPI$'s of F. It is denoted by $C(F/D)$.

As was mentioned before, the identification of the D core is usually a necessary step in the process of simplifying a truth function. In the following we shall state and prove a method for identifying the $CDPI$'s and give several examples for illustration.

Let F and D be truth functions of the letters $x_1, x_2, \cdots, x_n$. Let $G=g_1+g_2+\cdots+g_s$ be in $A(F/D)$, where the g_i's are DPI's of F. Suppose that $g_1=x_1x_2 \cdots x_r$, where $r \geq 1$. Any term f of the x_i's and $\bar{x}_i$'s must be of the following form:

$$f = \prod_{i \in A} x_i \prod_{i \in A'} \bar{x}_i \prod_{j \in B} x_j^{u_j}, \tag{1}$$

where A and A' are subsets of the set consisting of $1, 2, \cdots, r$; A and A' have no common elements; B is a subset of the set containing $r+1, r+2, \cdots, n$; and $x_j^{u_j}$ is either x_j or $\bar{x}_j$. (If A or A' or B is empty, then the corresponding factor is 1.) Define

$$f' = \prod_{j \in B} x_j^{u_j}, \qquad \text{if } A' \text{ has at most one element,}$$
$$= 0, \qquad \text{otherwise.} \tag{2}$$

Let $D=d_1+d_2+\cdots+d_t$ where the d_i's are terms.

The following theorem provides a means for identifying the D core of F. It is a generalization of Quine's[1] Theorem IX. (Quine did not consider don't care conditions and assumed that F is irredundant. Further, he was not concerned with simplifying truth functions.)

Theorem: $g_1 \in C(F/D)$ if and only if

$$G' + D' \neq 1, \tag{3}$$

where

$$G' = \sum_{k=2}^{s} g_k', \quad \text{and} \quad D' = \sum_{k=1}^{t} d_k'.$$

Proof: Suppose that $g_1 \in C(F/D)$. Let the DPI's of F not included in G be $g_{s+1}, g_{s+2}, \cdots, g_m$. Let us say that there exists an n term (a term with n literals) $h=g_1g$, where g is a term of the x_j's and $\bar{x}_j$'s, $j=r+1, \cdots, n$, ($g \equiv 1$ if $r=n$) such that, when $h=1$,

$$D + G_1 = 0, \quad \text{where} \quad G_1 = \sum_{k=2}^{m} g_k. \tag{4}$$

[4] T. H. Mott, Jr., "An Algorithm for Determining Minimal Normal Forms of an Incomplete Truth Function," presented at AIEE Fall General Meeting, Chicago, Ill.; October, 1959.

[5] R. J. Nelson, "Weak simplest normal truth functions," *J. Symbolic Logic*, vol. 20, pp. 232–234; September, 1955.

[6] Quine, *op. cit*, Theorems IV–VII.

This may be shown in the following way. When $D=0$, either: 1) $g_1 \equiv 0$, or 2) $g_1 \not\equiv 0$, and $g_1 \rightarrow G_1$, or 3) $g_1 \not\equiv 0$, but g_1 does not imply G_1. If 1) were true, then g_1 could be deleted from every $H \in A(F/D)$ which contains g_1 and the resulting form would still belong to $D(F/D)$. Hence $g_1 \notin C(F/D)$, which is a contradiction. If 2) were true, then when $D=0$, $G_1 = g_1 + G_1 = F + D = F$, or $G_1 \in A(F/D)$. But G_1 does not contain g_1, which again contradicts the assumption that $g_1 \in C(F/D)$. Therefore 3) must be true, and so is (4).

Now take any g_k, $2 \leqq k \leqq m$. If the corresponding A, defined in (1) is empty, then following (4), $h=1 \rightarrow g_k' = g_k = 0$. If A' has one element p, then $g_1 * g_k$ (the consensus of g_1 and g_k) exists and

$$g_1 * g_k = g_1' \cdot g_k',$$

where

$$g_1' = \prod_{\substack{i=1 \\ i \neq p}}^{r} x_i.$$

It is easy to show that $g_1 * g_k \rightarrow g_1 + g_k \rightarrow F + D$. So $g_1 * g_k$ is an implicant of $F+D$. Hence there exists a g_j which is a PI of $F+D$ and is subsumed by $g_1 * g_k$. Obviously $g_j = f_1 f_2$ where f_1 is a term subsumed by g_1', and f_2 contains at least one literal and is subsumed by g_k'.[7] It follows from (4) that when $h=1$, $f_2 = g_j = 0$, and $g_k' = 0$. Therefore $h=1 \rightarrow G' = 0$. Similarly it can be seen that $h=1 \rightarrow D' = 0$.

We now show that if $G' + D' \not\equiv 1$, then $g_1 \in C(F/D)$. Suppose that, on the contrary, $g_1 \notin C(F/D)$. Then there exists an H in $A(F/D)$ which does not contain g_1. Therefore

$$G_1 = \sum_{k=2}^{m} g_k = F(D).$$

Let[8]

$$g = \prod_{j=r+1}^{n} x_j^{u_j},$$

where $x_j^{u_j}$ is either x_j or $\bar{x}_j$, such that $g=1 \rightarrow D' = 0$ and $h = g_1 g$. When $h=1$, $D=0$. (Those terms in D which contain at least one $\bar{x}_i$, $i=1, \cdots, r$ vanish because $g_1 = 1$, and those d_k containing no $\bar{x}_i$, $i=1, \cdots, r$ vanish because $g_1 = 1 \rightarrow d_k = d_k'$ and $g=1 \rightarrow d_k' = 0$.) Further, $g_1 = 1 \rightarrow F + D = 1$. Since $G_1 = F(D)$, $G_1 = 1$ when $h=1$. Therefore h subsumes a $g_k = f_1 f$ in G_1, where f_1 is properly subsumed by g_i and f contains at least one letter and is subsumed by g. Let

$$g_k^* = \left(\prod_{\substack{i=1 \\ i \neq p}}^{r} x_i \right) \bar{x}_p g,$$

where $1 \leqq p \leqq r$ and x_p is not in f_1. Then $g_k^* = 1 \rightarrow g_k = 1 \rightarrow F + D = 1$. But $g_k^* = 1 \rightarrow D = 0$. (Those terms in D which contain two or more $\bar{x}_i$'s, $i=1, \cdots, r$, are 0 because

$$\prod_{\substack{i=1 \\ i \neq p}}^{r} x_i = 1,$$

and the remaining terms d_k in D are 0 because $g_k^* = 1 \rightarrow g = 1 \rightarrow d_k' = 0 \rightarrow d_k = 0$.) Since $G = F(D)$, $g_k^* = 1 \rightarrow G = 1$. Now g_k^* is an n term. Therefore g_k^* subsumes a term $g_j (\neq g_1)$ in G, and the corresponding g_j' is subsumed by g. Hence $g=1$ and $D'=0$ imply that $G'=1$, or $G' + D' \equiv 1$, which is a contradiction. Therefore $g_1 \in C(F/D)$.

Example 1:

Let[9] $F = \bar{a}b\bar{c} + \bar{a}\bar{c}d + \bar{b}\bar{d}$, and $D = ad + ac + cd + \bar{a}\bar{b}\bar{d}$. First we must test whether the terms in F are DPI's of F. This may be done systematically as follows. Take, for example, $\bar{a}b\bar{c}$. Change the rightmost literal to its negation and test whether the new term is a DI of F. In this case, the new term is $\bar{a}bc$. When $\bar{a}bc=1$, $D=d$; when $\bar{a}bc=1$ and $D=d=0$, $F=0$. Therefore $\bar{a}bc$ is not a DI of F. Now change the second literal in $\bar{a}b\bar{c}$ to its negation. When $\bar{a}\bar{b}\bar{c}=1$, $D=\bar{d}$. when $\bar{a}\bar{b}\bar{c}=1$ and $D=\bar{d}=0$, $F=1$. Therefore $\bar{a}\bar{b}\bar{c}$ is a DI of F and so is $\bar{a}\bar{c}$. Now change a to its negation and repeat the same procedure. We find that $a\bar{c}$ is not a DI, and hence $\bar{a}\bar{c}$ is a DPI of F. Now the second term $\bar{a}\bar{c}d$ in F may be eliminated. A similar method may be applied to the last term $\bar{b}\bar{d}$ and we find that $\bar{b}$ is a DPI of F. Thus $G = \bar{a}\bar{c} + \bar{b} = F(D)$ and the terms in G are DPI's of F.

Let $g_1 = \bar{a}\bar{c}$ and $g_2 = \bar{b}$. With respect to g_1, we see that $G' + D' = \bar{b} + d + d + \bar{b}\bar{d} = \bar{b} + d \not\equiv 1$. Hence, g_1 is a $CDPI$. With respect to g_2, $G' + D' = \bar{a}\bar{c} + ad + ac + cd + \bar{a}\bar{d} = 0$, when $a=1$, $c=0$ and $d=0$. Therefore g_2 is also a $CDPI$. One may have noticed that in the above example, F and D have a common implicant $\bar{a}\bar{b}\bar{d}$. In practice, F and D usually have no implicant in common. However, at least from a theoretical point of view, there is no harm in establishing a more general theorem without such a restriction.

Corollaries

The following corollary is a generalization of a method due to Ghazala[10] for identifying core prime implicants of a truth function when all prime implicants of the function are given, but no don't care condition is assumed.

Corollary 1: If G in the Theorem contains all the PI's of $F+D$ (*i.e.*, $s=m$), then $g_1 \in C(F/D)$ if and only if $G'' + D'' \not\equiv 1$, where

[7] If $g_k' \equiv 1$, then g_1 properly subsumes $g_1 * g_k$. Hence, g_1 is not a PI of $F+D$.

[8] For $r=n$, the proof is simpler. If $g_1 = 1 \rightarrow D = 1$, then g_1 subsumes a d_k and $d_k' \equiv 1$. This contradicts that $G' + D' \not\equiv 1$. Hence $g_1 \rightarrow D = 0$. Since $G_1 = F(D)$, $g_1 = 1 \rightarrow G_1 = 1$, or g_1 subsumes a g_0—another contradiction.

[9] This example is taken from M. Phister, Jr., "Logical Design of Digital Computers," John Wiley and Sons, Inc., New York, N. Y., pp. 102—103; 1959.

[10] Ghazala, *op. cit.*, p. 173.

$$G'' = \sum_{k=2}^{m} g_k'' \quad \text{and} \quad g_k'' = g_k'$$

if the corresponding set A', defined in (1), is empty, and $g_k''=0$ if otherwise, and similarly for D'' and d_k''.

Proof: If $g_1 \in C(F/D)$, it follows from the Theorem that $G'+D' \not\equiv 1$; hence $G''+D'' \not\equiv 1$. Now suppose that $G''+D'' \not\equiv 1$, but $g_1 \notin C(F/D)$. Then, according to the Theorem, $G'+D' \equiv 1$. This means for any assignment of values to the x_j's, $j=r+1, r+2, \cdots, n$, there exists a g_k' or d_k' whose corresponding value is 1. If the corresponding set A', defined in (1), is empty, then $g_k''=g_k'$ or $d_k''=d_k'$ are also 1. Therefore $G''+D''=1$. Now suppose that $g_k'=1$ and A' have one element. In such a case

$$g_1 * g_k = \left(\prod_{\substack{i=1 \\ i \neq p}}^{r} x_i \right) g_k'$$

exists, where $1 \leqq p \leqq r$. Since $g_1 * g_k \rightarrow F+D$, $g_1 * g_k$ subsumes a PI g_j of $F+D$. Now g_j'' is subsumed by g_k'. Therefore $g_k''=1$ and $G''=1$. Similarly if $d_k'=1$ and the corresponding A' has one element, then $G''=1$. Thus $G''+D'' \equiv 1$, which is a contradiction.

Example 2:

Let $F=ab+\bar{b}c+acd+bc\bar{d}$ and $D=a\bar{b}d+\bar{a}c$. The prime implicants of $F+D$ may be obtained by, for example, a method of Quine.[6] They are $g_1=ab$, $g_2=ad$, and $g_3=c$. Corresponding to g_1, g_2 and g_3, respectively,

$$\begin{aligned} G'' + D'' &= d + c \not\equiv 1, \\ G'' + D'' &= b + c + \bar{b} \equiv 1, \\ G'' + D'' &= ab + ad + a\bar{b}d + \bar{a} \\ &= 0, \quad \text{if} \quad a\bar{b}\bar{d} = 1. \end{aligned}$$

Therefore g_1 and g_3 are $CDPI$, but g_2 is not.

In (2), we defined f' for any term f with respect to a given term g_1. We now define $f^*=g_1 f'$, and $f^{**}=g_1 f''$, where $f''=f'$ if the corresponding set A' is empty, and $f'' \equiv 0$, if otherwise. Then:

Corollary 1a): Using the notations of Corollary 1, $g_1 \in C(F/D)$ if and only if g_1 cannot be obtained through repeated application of the consensus operation to the g_k^{**}'s, $k=2, \cdots, m$.

The above corollary is equivalent to a Theorem of Roth.[11] The proof is very similar to that of Corollary 2 and is therefore omitted.

Corollary 2: Using the notations of the Theorem, $g_1 \in C(F/D)$ if and only if g_1 cannot be obtained through repeated applications of the consensus operation to the g_k^*'s, $k=2, \cdots, s$, and d_k^*'s, $k=1, \cdots, t$.

Proof: Suppose that g_1 may be obtained through repeated applications of the consensus operation to g_2^*, $g_3^*, \cdots, g_p^*$, and $d_1^*, d_2^*, \cdots, d_q^*$ (in a certain order).

Then $g_1 \rightarrow g_2^* + \cdots + g_p^* + d_1^* + \cdots + d_q^*$. Since $f^*=f'$ when $g_1=1$, $g_1 \rightarrow G'+D'$. But $G'+D'$ is not a truth function of the x_i's, $i=1, \cdots, r$. Therefore $G'+D' \equiv 1$ and $g_1 \notin C(F/D)$. Further, if $g_1 \notin C(F/D)$, then

$$g_1 = 1 \rightarrow G^* + D^* = \sum_{k=2}^{s} g_k^* + \sum_{k=1}^{t} d_k^* = G' + D' = 1.$$

Meanwhile $G^*+D^*=1 \rightarrow g_k^*=1$ or $d_k^*=1$, for some k. Hence $g_1=1$ and $F+D=1$. Therefore g_1 is a PI of G^*+D^*. It follows from Quine's[6] Theorems IV–VII that g_1 may be derived from repeated applications of the consensus operation to the g_k^*'s and d_k^*'s.

The above corollary is slightly different from a result of Roth,[12] where he required that the d_k's be PI's of D. A disadvantage of this method for identifying the core is that in some cases a large number of terms may be generated through repeated applications of the consensus operation to the g^*'s and d^*'s. It is quite possible, however, that in other instances his method is more time-saving than ours.

Based on his methods for identifying the $CDPI$'s, Roth[2] also suggested two methods for simplifying truth functions, known as the Extraction Algorithms. According to these algorithms, the simplification process is carried out in an inductive fashion. At each stage, the $CDPI$'s are identified and extracted. When the process reaches a stage where no DPI under consideration belongs to the core, a partial ordering is introduced, the nonmaximal terms are eliminated, and a branching operation is applied to the remaining terms. Obviously our methods for identifying the $CDPI$'s may be used as substitutes for his. (We omit the details.)

Example 3:

Take Example 1. Corresponding to g_1 and g_2 respectively,

$$G^* + D^* = \bar{a}\bar{b}\bar{c} + \bar{a}\bar{c}d, \tag{5}$$

$$G^* + D^* = \bar{a}\bar{b}\bar{c} + a\bar{b}d + a\bar{b}c + \bar{b}cd + \bar{a}\bar{b}\bar{d}. \tag{6}$$

No consensus exists for the terms that appeared in the right-hand side of (5). Hence g_1 is a $CDPI$. Applying the consensus operation to the terms in the right-hand side of (6), we obtain the following terms: $\bar{a}\bar{b}$, $\bar{b}c$, and $\bar{b}d$. None of them is g_2; therefore, g_2 is a $CDPI$.

[11] Roth, *op. cit.*, p. 65.

[12] *Ibid.*, p. 71.

Reducing Computing Time for Synchronous Binary Division*

ROY G. SALTMAN†, MEMBER, IRE

Summary—The computing time for binary division is shortened by performing division, radix 2^p on the binary operands, where p is a positive integer. Each quotient digit radix 2^p is computed in almost the same time required to determine a binary quotient digit. Therefore, computing time is reduced by approximately the factor p over conventional binary division. The method is most useful for synchronous machines but can be applied to either serial or parallel operation.

The theory of nonrestoring division in any integral radix r is discussed. Each quotient digit is considered as the sum of two recursive variables a_k and b_k, whose values depend on the divisor multiplier and relative signs of the partial remainders. The divisor multiplier is limited to odd integers in order to determine the quotient digit unambiguously. Using a_k and b_k, a single recursive equation combining all sign conditions is derived. This permits the derivation of the correct round-off procedure and shows that binary nonrestoring division is a particular case of nonrestoring division, radix r.

An arrangement of components for a serial computer and a sample division for radix four are given.

Introduction

In a binary digital computer, a nonrestoring method of division is often employed. This method allows the direct employment of operands of either sign and generates the correctly-signed quotient without complementation.

However, computation time is long because conventional nonrestoring division consists of a sequence of addition and subtraction operations, and each operation cannot be completed until the results of the previous operation have been determined. For word lengths of n bits, the time used for a conventional division is n or more times the addition or subtraction time, and in a serial computer, can be $n/2$ times the multiplication time if the multiplication unit is of the serial-parallel type described by Richards.[1]

Nonrestoring binary division has been described by Burks, Goldstine, and von Neumann,[2] and by Booth and Booth.[3] In this method, one binary quotient bit is determined in each iteration of a recursive process. The recursive equation for the kth iteration generates a new partial remainder X_k as a function of the present partial remainder X_{k-1} and the divisor D. The dividend X_0 is the remainder before the first iteration. The recursive equation includes either a subtraction or an addition depending on whether the relative signs of X_{k-1} and D are the same or different. The relationships are:

$$X_k = 2X_{k-1} - D, \qquad \operatorname{sgn} X_{k-1} = \operatorname{sgn} D \tag{1a}$$

$$X_k = 2X_{k-1} + D, \qquad \operatorname{sgn} X_{k-1} \neq \operatorname{sgn} D. \tag{1b}$$

The quotient bit determined by the recursive process is unity or zero according to whether the relative signs of X_k and D are the same or different, respectively.

It will be shown that nonrestoring division can be extended to any number system of radix r where r is a positive integer larger than unity. In particular, nonrestoring division is applicable to number systems of radix $r = 2^p$. This is useful because binary numbers may be considered to be binary-coded numbers of radix 2^p without altering their appearance. The change in radix is accomplished by identifying each digit radix 2^p with each corresponding group of p successive bits starting from the binary point. Therefore, nonrestoring division, radix 2^p, can be performed on binary numbers directly. Furthermore, radix 2^p division can be mechanized so that each quotient digit consisting of p bits is computed in almost the same time required to find a single binary quotient bit in nonrestoring binary division. With this procedure, computation time of division is reduced by approximately the factor p.

The mechanization of nonrestoring division, radix 2^p, for saving computing time is applicable to either serial or parallel fixed-point operation. It is particularly useful for synchronous machines, since computation time of the process depends only on the lengths of the variables involved, and not upon their particular values. For asynchronous machines in which a direct left shift is available, the methods described by Lehman[4] and MacSorly[5] for reducing computation time may also be employed. A new class of division methods useful for floating-point arithmetic has been described by Robertson.[6]

* Received by the PGEC, July 14, 1960; revised manuscript received, January 30, 1961.

† Surface Armament Div., Sperry Gyroscope Co., Div. of Sperry Rand Corp., Great Neck, L. I., N. Y.

[1] R. K. Richards, "Arithmetic Operations in Digital Computers," D. Van Nostrand and Co., Inc., Princeton, N. J., pp. 155–157; 1953.

[2] A. W. Burks, H. H. Goldstine, and J. von Neumann, "A Preliminary Discussion of the Logical Design of an Electronic Computing Instrument," The Inst. for Advanced Study, Princeton, N. J., pt. 1, vol. 1, pp. 22–26; 1946.

[3] A. D. Booth and K. H. V. Booth, "Automatic Digital Calculators," Academic Press, Inc., New York, N. Y., 2nd ed., pp. 48–50; 1956.

[4] M. Lehman, "Short-cut multiplication and division in automatic binary digital computers," PROC. IRE, vol. 46, pp. 496–504; September, 1958.

[5] O. L. MacSorly, "High-speed arithmetic in binary computers," PROC. IRE, vol. 49, pp. 67–91; January, 1961.

[6] J. E. Robertson, "A new class of digital division methods," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-7, pp. 218–222; September, 1958.

Nonrestoring Division, Radix r

In the nonrestoring division procedure to be described, the quotient Q is a fixed-point number having the magnitude limitations:

$$-1 \leq Q < 1. \tag{2}$$

The quotient is represented by a sign digit q_0, plus n digits of integral radix r to the right of the radix point. The kth digit to the right is q_k where q_k is a non-negative integer less than r. If Q is zero or positive, q_0 is zero. If Q is negative, it is represented in r's-complement form as $r+Q$, and q_0 is $r-1$.

In the division process in general, the quotient Q plus a final remainder equals X_0/D. The final remainder may be zero, and the limitations on dividend and divisor using fractional operands are:

$$-1 \leq X_0/D < 1, \qquad 0 < D < 1 \tag{3a}$$

$$-1 < X_0/D \leq 1, \qquad -1 \leq D < 0. \tag{3b}$$

If D is positive definite as in (3a), the limits of X_0/D are the same as that given to Q in (2). If D is negative definite as in (3b), the division process to be described requires that X_0 be allowed to equal D. Then Q can be unity which is not permitted in (2). However, in this case the calculated value of Q is $1-r^{-n}$ which is permissible and correct to the resolution of Q. The divisor is not permitted to be zero.

If all iterations are identical in form, the limitations given to X_0/D must also be applied to X_k/D, where X_k is the kth partial remainder. Therefore:

$$-1 \leq X_k/D < 1, \qquad 0 < D < 1 \tag{4a}$$

$$-1 < X_k/D \leq 1, \qquad -1 \leq D < 0. \tag{4b}$$

In pencil-and-paper division, in which all variables are positive, the recursive equation defining the quotient digit q_k in radix r is:

$$X_k = rX_{k-1} - q_kD, \qquad 0 \leq q_k < r$$
$$0 \leq X_k/D < 1. \tag{5}$$

In (5) q_k and X_k are unknowns. Since q_k is integral, possible values of X_k differ by multiples of D. There is only one value of q_k which determines X_k to be in the range $0 \leq X_k/D < 1$.

In nonrestoring division, D, X_{k-1}, and X_k may have either sign independently, but q_k is defined as in (5) only when all variables are positive. For variables of either sign, the recursive relations are:

$$X_k = rX_{k-1} - m_kD, \qquad \text{sgn } X_{k-1} = \text{sgn } D \tag{6a}$$

$$X_k = rX_{k-1} + m_kD, \qquad \text{sgn } X_{k-1} \neq \text{sgn } D. \tag{6b}$$

In (6), m_k is a non-negative integer, and the selection of addition or subtraction of m_kD is such that X_k tends toward zero. Integer m_k is selected so that the magnitude limitations given in (4) are satisfied. Possible values of X_k differ by multiples of D, and exactly two possible values of X_k, one positive and one negative, fall in the range given in either (4a) or (4b). In either case, X_k can fall between $-|D|$ and $|D| - r^{-n}$ inclusive. In the lower limit, allowable values of X_k are $-|D|$ and zero (which is considered positive). In the upper limit, allowable values are $-r^{-n}$ and $|D| - r^{-n}$.

The two acceptable values of X_k must be generated by values of m_k which differ by unity. One of the possible values of m_k must be odd and the other even. If m_k is limited to odd integers in the range unity to r inclusive where r may be even or odd, then only one value of X_k can be found which satisfies (4) and any ambiguity is resolved.

The desired quotient digit q_k is a variable function of m_k depending on the relative signs of D, X_{k-1}, and X_k. There are four situations to be considered, established by whether the signs of D and X_{k-1} are the same or different and whether the signs of D and X_k are the same or different. The relationships of q_k to m_k, obtained as a result of a derivation given in the Appendix, are:

$$q_k = m_k, \qquad \text{sgn } X_{k-1} = \text{sgn } D, \quad \text{sgn } X_k = \text{sgn } D \tag{7a}$$

$$q_k = m_k - 1, \qquad \text{sgn } X_{k-1} = \text{sgn } D, \quad \text{sgn } X_k \neq \text{sgn } D \tag{7b}$$

$$q_k = r - m_k, \qquad \text{sgn } X_{k-1} \neq \text{sgn } D, \quad \text{sgn } X_k = \text{sgn } D \tag{7c}$$

$$q_k = r - m_k - 1, \qquad \text{sgn } X_{k-1} \neq \text{sgn } D, \quad \text{sgn } X_k \neq \text{sgn } D. \tag{7d}$$

The four cases given in (7) can be combined into a single recursive equation by defining q_k in terms of two variables a_k and b_k, such that:

$$q_k = a_k + b_k. \tag{8}$$

Let a_k be defined as follows:

$$a_k = m_k - 1, \qquad \text{sgn } X_{k-1} = \text{sgn } D \tag{9a}$$

$$a_k = r - m_k - 1, \qquad \text{sgn } X_{k-1} \neq \text{sgn } D. \tag{9b}$$

Also, b_k is defined as:

$$b_k = 1, \qquad \text{sgn } X_k = \text{sgn } D \tag{10a}$$

$$b_k = 0, \qquad \text{sgn } X_k \neq \text{sgn } D. \tag{10b}$$

Eqs. (7) and (8) define q_k in terms of different variables. If the two definitions are compared for similar sign conditions, it is seen that they are consistent.

The single recursive equation combining all conditions is:

$$X_k = rX_{k-1} - (rb_{k-1} + a_k - r + 1)D. \tag{11}$$

Note that a_k and b_{k-1}, which appear in (11), depend only on the relative signs of X_{k-1} and D. If values of a_k and b_{k-1} for particular sign conditions are substituted, (11) reduces to the corresponding (6a) or (6b).

The recursive process, from which the definitions of q_k have been derived, generates an infinite sequence of quotient digits. This is necessary since the dividend and divisor are, in general, incommensurable. However, the

truncation of the recursive process after the determination of n digits beyond the radix point requires some form of round-off of the quotient. This implies that the least significant digit q_k cannot be defined as in (7) or (8). Similarly, no suitable definition of the sign digit q_0 has been made, since the kth iteration determines q_k and there is no iteration prior to the first.

In order to determine the digit values in a quotient of n digits plus sign, the ratio X_0/D is derived by combining the first n iterations of (11). If in (11), $k=n$, a substitution of (11) with $k=n-1$ can be made for X_{n-1} to determine X_n in terms of X_{n-2}. After $n-2$ similar substitutions, X_n is determined in terms of X_0.

This final relationship is:

$$X_n = r^n X_0 - D \sum_{k=1}^{n} r^{n-k}(rb_{k-1} + a_k - r + 1). \quad (12)$$

After dividing both sides of (12) by $r^n D$ and rearranging, the following is obtained:

$$\frac{X_0}{D} = \frac{X_n}{r^n D} + \sum_{k=1}^{n} \frac{a_k}{r^k} + \sum_{k=1}^{n} \frac{b_{k-1}}{r^{k-1}} + \sum_{k=1}^{n} \left(\frac{1}{r^k} - \frac{1}{r^{k-1}} \right). \quad (13)$$

Eq. (13) can be simplified by noting that:

$$\sum_{k=1}^{n} \frac{a_k}{r^k} + \sum_{k=1}^{n} \frac{b_{k-1}}{r^{k-1}} = b_0 + \sum_{k=1}^{n-1} \frac{(a_k + b_k)}{r^k} + \frac{a_n}{r^n}. \quad (14)$$

Also:

$$\sum_{k=1}^{n} \left(\frac{1}{r^k} - \frac{1}{r^{k-1}} \right) = -1 + \frac{1}{r^n}. \quad (15)$$

Substituting (14) and (15) into (13) yields the desired ratio in terms of successive powers of $1/r$:

$$\frac{X_0}{D} = b_0 - 1 + \sum_{k=1}^{n-1} \frac{(a_k + b_k)}{r^k} + \frac{(a_n + 1)}{r^n} + \frac{X_n}{r^n D}. \quad (16)$$

The final remainder in (16) is $X_n/r^n D$, so that the quotient includes everything else on the right-hand side. If X_0 and D have the same sign, then b_0 is unity, and

$$0 \leq Q = \sum_{k=1}^{n-1} \frac{(a_k + b_k)}{r^k} + \frac{(a_n + 1)}{r^n} < 1. \quad (17)$$

If X_0 and D have opposite signs, b_0 is zero and the value of Q is given by:

$$0 > Q = -1 + \sum_{k=1}^{n-1} \frac{(a_k + b_k)}{r^k} + \frac{(a_n + 1)}{r^n} \geq -1. \quad (18)$$

The representation of Q in r's-complement form when Q is negative is as $r+Q$ where Q is given in (18). Therefore:

$$r + Q = r - 1 + \sum_{k=1}^{n-1} \frac{(a_k + b_k)}{r^k} + \frac{(a_n + 1)}{r^n}. \quad (19)$$

From (16)–(19), it is clear that the various digits of Q can be defined as follows:

$$q_0 = (r + b_0 - 1) \text{ modulo } r \quad (20a)$$

$$q_k = a_k + b_k, \qquad k = 1, 2, \cdots, n-1 \quad (20b)$$

$$q_n = a_n + 1. \quad (20c)$$

Simplifications in Radix 2^p for Binary Operands

In applying division radix 2^p, the apparent additions or subtractions which are required to form the quotient digits in (20) are eliminated. For example, since the desired quotient is ultimately a binary number, the sign digit q_0 must have either the value zero or unity. Thus, in (20a), r is two, and q_0 is $(1+b_0)$ mod 2. The addition, mod 2, of unity to b_0 is effectively the same as inverting b_0.

Addition is also eliminated in the formation of the other quotient digits. In any number system in which r is even (as it must be for radix 2^p), m_k is an odd integer between unity and $r-1$ inclusive. It can be seen from the definitions given to a_k in (9) that, in this case, a_k is an even integer between zero and $r-2$ inclusive. Assuming bits with binary weights, a binary-coded digit, radix 2^p is even if the right-most bit is zero. Therefore, the value of a_k affects only the left-most $p-1$ bits of q_k, and the addition of b_k affects only the right-most bit of q_k. Similarly, the left-most $p-1$ bits of q_n are determined solely by a_n, and the addition of unity to form q_n is equivalent to setting the right-most bit (b_n) to unity.

For division, radix 2^p, b_n is a binary bit of weight 2^{-np}. Setting the least significant bit to unity for round-off is called "stuffing." For a variable which is positive and negative with equal frequency, and whose true value is an infinite sequence of digits, stuffing provides an unbiased round-off and a standard deviation of $1/\sqrt{3}$ times the value of the bit stuffed. This has been shown by Burks, Goldstine, and von Neumann.[7]

Then the computer steps for determining a binary-coded quotient of n digits, radix 2^p and a binary sign digit are as follows:

1) Compare the signs of X_{k-1} and D. If they are the same, set b_{k-1} to unity. If they are different, set b_{k-1} to zero.
2) Form $2^p X_{k-1}$ from X_{k-1}.
3a) If b_{k-1} is unity, subtract $m_k D$ from $2^p X_{k-1}$ forming X_k, where m_k is the odd positive integer such that $-|D| \leq X_k < |D|$. Set $a_k = m_k - 1$.
3b) If b_{k-1} is zero, add $m_k D$ to $2^p X_{k-1}$ forming X_k, where m_k is the odd positive integer such that $-|D| \leq X_k < |D|$. Set $a_k = 2^p - m_k - 1$.
4) Perform the above steps for all values of k from 1 to n. Then invert b_0 and set b_n to unity.

[7] Burks, *et al.*, *op. cit.*, pp. 19–22.

Mechanization of Division Radix 2^p

In order to save maximum time in the division process, all odd and even integral multiples of the divisor up to the (2^p-1)st multiple must be formed simultaneously. In a serial computer, the multiples need not be stored, but can be regenerated in each iteration by causing the divisor to serially feed into combinations of doublers (single-bit delays) and adders. As the 2^p-1 multiples are serially formed, each is serially added to or subtracted from 2^pX_{k-1}, forming 2^p-1 potential values of X_k.

In a parallel computer, in order to save maximum time, all multiples of the divisor must be formed first before the iterative process is begun. The stages of the register holding the divisor can statically control the inputs to several parallel adders so that the outputs of the adders form all necessary multiples of the divisor in parallel. Each group of output lines forming a divisor multiple can similarly control one of the inputs to a parallel adder/subtracter. Each adder/subtracter receives its second parallel input from a register holding 2^pX_{k-1}, and the outputs of the adder/subtracters are the potential values of X_k.

In either serial or parallel operation, the actual value of X_k is selected from among the potential values formed from odd multiples of the divisor. However, the final determination cannot be made until the signs of all potential values are known. In a serial computer, storage must be provided for all potential values formed from odd-divisor multiples, since the sign digit is determined last in addition or subtraction. However, only the sign digits of potential X_k formed from even-divisor multiples must be stored. In a parallel computer, if all potential X_k are available simultaneously as static outputs of the adder/subtracters, the determination of the actual X_k can be made first and then only that value transferred to storage.

In either type of operation, the actual value of X_k is determined solely from the sign digits of the potential values. Each sign digit of the potential values consists of p bits, radix 2^p, and can be any integer between zero and 2^p-1 inclusive. However, the actual value of X_k must have a sign digit of either zero or 2^p-1 but no other value, and either of these values or both must occur as sign digits among the potential values. The potential values of X_k form a monotonic function plotted against increasing multiples of the divisor, and therefore there can be at most only one change in sign among them. The change in sign, if it occurs, will fall between two potential X_k having sign digits of zero and 2^p-1, respectively. Whichever of these potential values was formed with an odd divisor multiple is selected as the actual X_k, and that odd divisor multiple is selected as m_k.

If the potential values of X_k have no change in sign, either zero or 2^p-1 but not both must occur as sign digits of potential X_k. The acceptable sign digit which occurs must be the same or different than the sign of X_{k-1} and the value of m_k is determined accordingly. If the sign digit of X_{k-1} also occurs among the potential X_k, m_k is 2^p-1, whereas if the sign of X_{k-1} does not occur as a sign digit among the potential X_k, m_k is unity. The potential X_k formed with the determined m_k is selected as the actual X_k. To illustrate the process a sample nonrestoring division, radix four, is shown in Fig. 1.

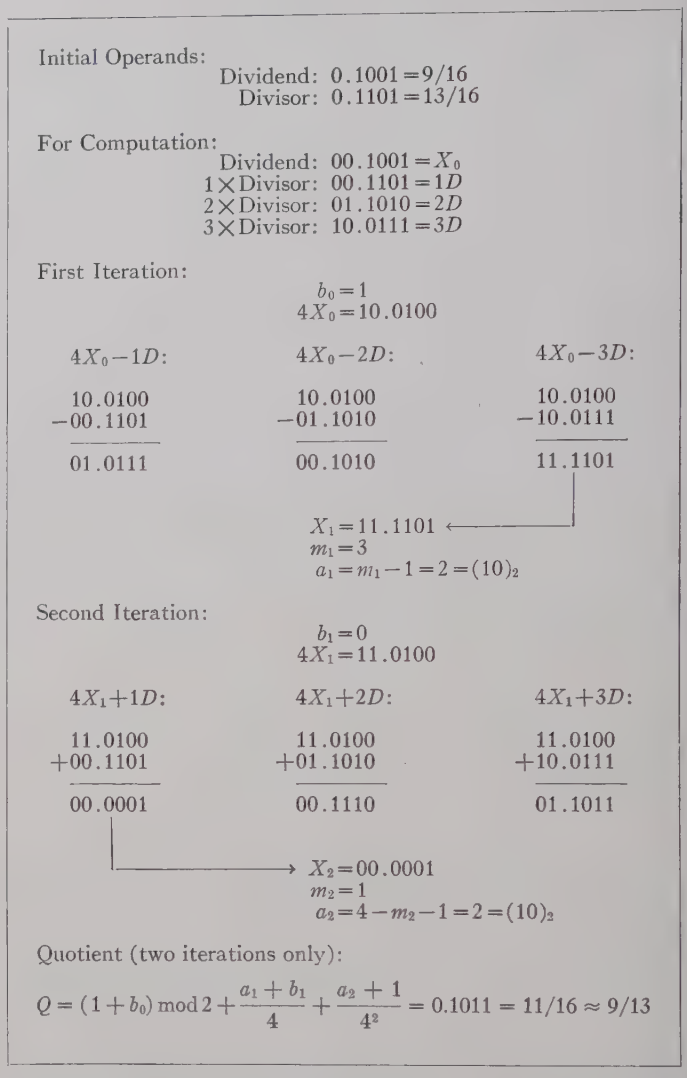


Fig. 1—Sample nonrestoring division, radix 4, with binary operands.

An arrangement of components for serial nonrestoring division, radix four, is shown in Fig. 2. Four full-length registers are required. These registers hold the divisor, the quotient, the potential X_k formed from the divisor itself (stored in R_1), and the potential X_k formed from the third multiple of the divisor (stored in R_3). The dividend is initially stored in R_1. Register R_2 consists of two bits and holds the sign digit, radix four, of the potential X_k formed from the second divisor multiple.

All resisters shift to the right only. In each iteration, the quotient is shifted left two bits by recirculation so

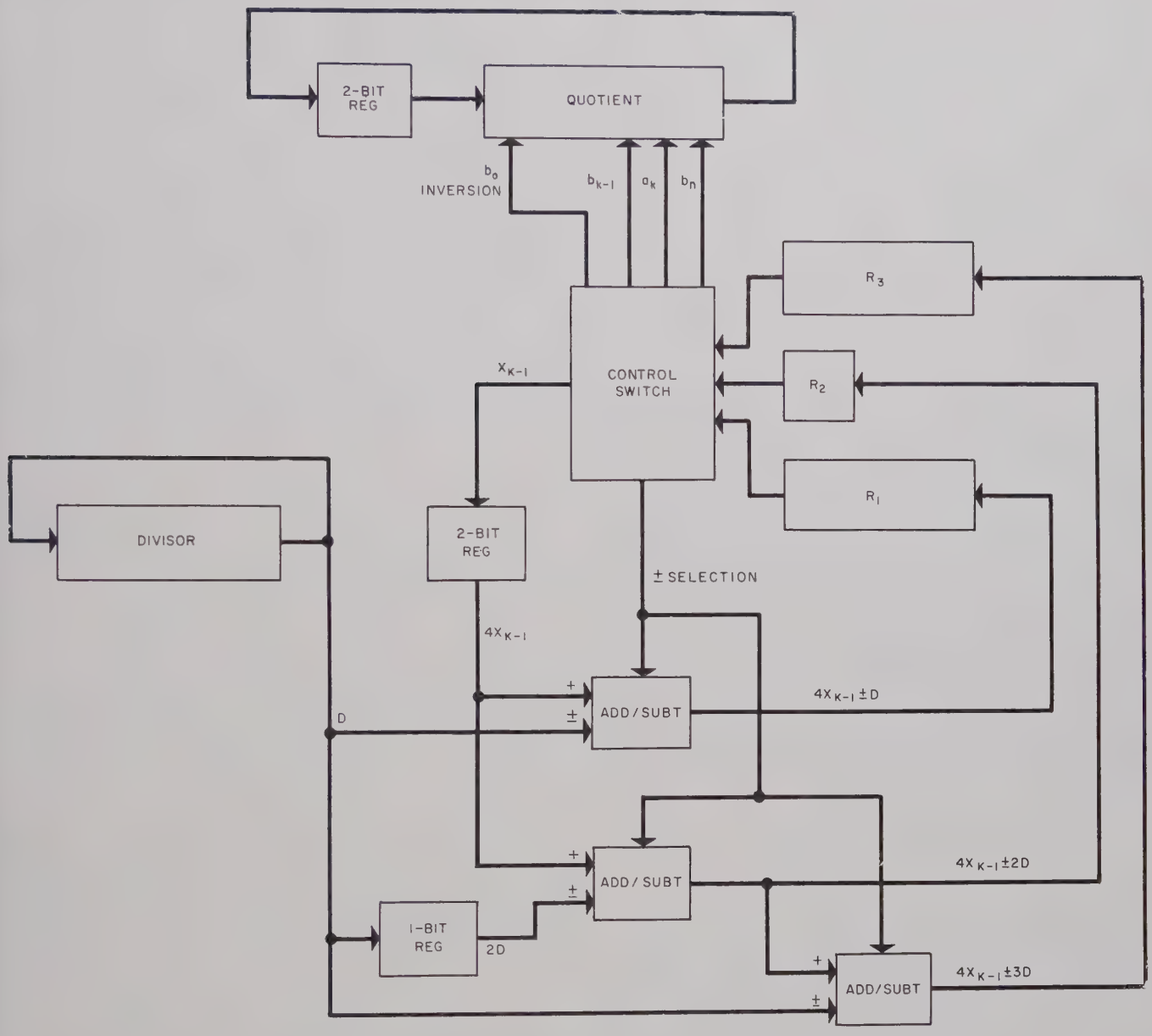


Fig. 2—Serial, nonrestoring division, radix 4, for a binary computer.

that b_{k-1} and a_k are always set into the same stages of the quotient register. To multiply X_{k-1} by four, its transmission is delayed two bits with respect to D. The purpose of the Control Switch is to select X_k, establish a_k and b_{k-1}, select addition or subtraction, and on the last iteration set b_n and invert b_0.

The extension of Fig. 2 to any division, radix 2^p, is clear. In general, 2^{p-1} full-length registers are required to hold potential X_k and 2^p-1 adder/subtracters are also required.

Conclusions

The nonrestoring binary division process described by Burks, Goldstine, and von Neumann is a particular example of general nonrestoring division for any integral radix r. This can be seen by considering the composite recursive equation (11) in the binary case. In this case $r=2$, $m_k=1$, $a_k=0$. With these substitutions for r and a_k, (11) reduces to either (1a) or (1b), according to the substitution of the possible values of b_{k-1}.

The development of nonrestoring division radix r is aided by the definition of the general quotient digit q_k as the sum of two recursive variables a_k and b_k. The definition of the quotient digit in this manner permits the sign digit to be a function of b_0 and makes possible a derivation of the proper round-off procedure for the least significant digit. The process of "quotient conversion" as stated by Robertson[6] is simplified. Furthermore, for an even radix r, the arithmetic sum of a_k and b_k can be replaced by a simple logical sum.

The divisor multiple m_k used in forming the new partial remainder X_k is limited to odd positive integers. This avoids ambiguity in the selection of X_k, and helps

limit the quantity of equipment required to store potential values of X_k in a serial mechanization of division radix 2^p for reducing computing time.

The reduction in computing time when performing division radix 2^p in a serial computer is less than factor p. The full reduction is not obtained because each iteration is lengthened by $p-1$ bits over the binary case to accommodate sign digits of length p in potential X_k. The extension of nonrestoring division to any nonbinary radix such as ten is clear.

Appendix

Relationship of q_k to m_k

Eq. (5) defines the quotient digit q_k for D, X_{k-1}, and X_k, having positive signs. Eq. (6) defines m_k subject to the limitations of (4). The relationships of q_k to m_k under different sign conditions are to be shown. In the derivations below, the inequalities between X_k and D or between X_{k-1} and D are given for D positive. If D is negative, the inequalities are the same, excepting that the equalities in all lower bounds are omitted and replaced by equalities in the upper bounds. All other equations remain the same. Similarly, (5) can also define q_k for D, X_{k-1}, and X_k having negative signs if the condition $0 \leq X_k/D < 1$ is replaced by $0 < X_k/D \leq 1$.

Case I: For D, X_{k-1}, and X_k having the same signs, (6a) defines m_k, and it reduces to (5) identically, subject to the inequality condition mentioned above. In this case, subtraction is employed and:

$$q_k = m_k, \qquad \begin{aligned} &\operatorname{sgn} X_{k-1} = \operatorname{sgn} D \\ &\operatorname{sgn} X_k \;\; = \operatorname{sgn} D. \end{aligned} \tag{21}$$

Case II: Again let D and X_{k-1} have the same sign, but let X_k and D have opposite signs. Then if $-1 \leq X_k/D < 0$, then $0 \leq X_k'/D < 1$ if:

$$X_k' = X_k + D. \tag{22}$$

Now (5) can be applied to D, X_{k-1} and X_k' as follows:

$$X_k' = rX_{k-1} - q_k D. \tag{23}$$

Then substituting for X_k' given in (23) into (22) and solving (22) for X_k:

$$X_k = rX_{k-1} - (q_k + 1)_{\operatorname{sgn} D}. \tag{24}$$

In this case, subtraction is employed and:

$$q_k = m_k - 1, \qquad \begin{aligned} &\operatorname{sgn} X_{k-1} = D \\ &\operatorname{sgn} X_k \;\; \neq \operatorname{sgn} D. \end{aligned} \tag{25}$$

Case III: Now let D and X_{k-1} have opposite signs where $-1 \leq X_{k-1}/D < 0$. Then $0 \leq X_{k-1}'/D < 1$ if:

$$X_{k-1}' = X_{k-1} + D. \tag{26}$$

Now (5) can be applied to D, X_{k-1}' and X_k as follows, if D and X_k have the same sign:

$$X_k = rX_{k-1}' - q_k D. \tag{27}$$

Now substituting for X_{k-1}' given in (26) into (27):

$$X_k = rX_{k-1} + (r - q_k)D. \tag{28}$$

In this case, in order for m_k to be a non-negative integer, addition is used to form X_k, and:

$$q_k = r - m_k, \qquad \begin{aligned} &\operatorname{sgn} X_{k-1} \neq \operatorname{sgn} D \\ &\operatorname{sgn} X_k \;\; = \operatorname{sgn} D. \end{aligned} \tag{29}$$

Case IV: Again let D and X_{k-1} have opposite signs and define X_{k-1}' as in Case III so that $0 \leq X_{k-1}'/D < 1$. Also let D and X_k have opposite signs. Now if $0 \leq X_k'/D < 1$, (5) can be applied to D, X_{k-1}' and X_k':

$$X_k' = rX_{k-1}' - q_k D. \tag{30}$$

However, the relationship of X_{k-1}' to X_{k-1} is the same as in (26) and the relationship of X_k' to X_k is the same as in (22).

Substituting (26) and (22) into (30) and solving for X_k:

$$X_k = rX_{k-1} + (r - q_k - 1)D. \tag{31}$$

In this case, addition is used, and

$$q_k = r - m_k - 1, \qquad \begin{aligned} &\operatorname{sgn} X_{k-1} \neq \operatorname{sgn} D \\ &\operatorname{sgn} X_k \;\; \neq \operatorname{sgn} D. \end{aligned} \tag{32}$$

The Philips Computer PASCAL*

H. J. HEIJN AND J. C. SELMAN†

Summary—**PASCAL is a binary parallel computer with a word length of 42 bits, a clock-pulse repetition time of $1\frac{1}{2}$ μsec, performing, on the average, 60,000 operations per second. Wired-in floating-point facilities are provided. Core storage is backed by a drum and by magnetic tape. There are modification versions for indexing as well as for stepping-up purposes. Special instructions include count and repeat instructions, jumps on the result of an earlier comparison, and two kinds of link instructions for facilitating the use of subroutines and interpretative programs. Transfer instructions enable a simultaneous bidirectional data flow between drum and cores or between tape and cores while computations are going on.**

Introduction

THE PAPER gives a brief description of the Philips Automatic Sequence CALculator, PASCAL.

The computer has been developed at Philips to serve two purposes: to gain field experience in the application of computer components and subassemblies, and to increase the firm's computing capacity for technical and scientific problems. PASCAL's development was started at the Research Laboratories by the authors in 1956 and from 1957 was worked out by a team of about seven with support of many others.

In the compromise between speed, simplicity of use, and circuitry, stress was laid on two points: circuits should be used efficiently and the internal machine code should be easy.

Several of PASCAL's features turned out to be in common with other computers. Some are new to the best of our knowledge, the most interesting ones being:

1) the speeding-up arrangements for the carry in the adder and the multiple-way switch,
2) the two versions for modification of instructions,
3) the count and repeat instructions and the two types of link instructions for going to subroutines,
4) the jump instructions on the result of an earlier comparison,
5) the address-interpretation bit and the indicator-word,
6) the simultaneous data transfer in both directions between core memory and drum or tape while computations go on.

The paper is divided into three sections: a description of the machine, a discussion of some of the programming features, and a description of constructional details.

* Received by the PGEC, January 30, 1961.
† Philips Res. Labs., N. V. Philips' Gloeilampenfabrieken, Eindhoven, The Netherlands.

I. Description of the Machine

A. The Words

PASCAL is a binary parallel computer having a 42-digit word length and working on a clock-pulse repetition rate of 660 kc. The time interval between two pulses of $1\frac{1}{2}$ μsec forms a basic period for the machine.

Words may represent *fixed-point numbers* of 41 digits +sign (negative numbers in one's complement notation), or *floating-point numbers* to the base 2. Here 33 digits+1 sign digit denote the mantissa (p) and 7 digits +sign digit specify the exponent part (q), the value of the representation then being $p \cdot 2^q$. Usually $|p|$ is normalized as $\frac{1}{2} \leq |p| < 1$. The range of these numbers thus is roughly between 10^{+38} and 10^{-38} with a precision of almost 10 decimals.

One 42-bit word stores *two instructions*, 0_1 and 0_2; 0_1 is the least significant half and is carried out first. The instructions are built up according to Fig. 1.

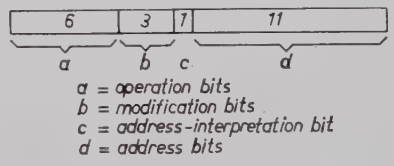


Fig. 1—Arrangement of the instructions.

The 11 address digits indicate 2048 possible addresses in the primary or working store. Of these 2016 refer to a ferrite-core store having a 6-μsec cycle time. The remaining 32 addresses (0–31) are divided as follows:

16 words (0–15) are on a plug board on the control desk and are meant for constants, initial orders, etc. Address 0, which is used for starting a program, even has special push buttons for quick operation.

8 words (addresses 16–23) are modification registers. These are full-instruction (21 bits) p-n-p-n flip-flop registers, which are selected by the 3 modification bits (see Fig. 1).

8 words (addresses 24–31) are special. Amongst these are the accumulator (A) and the multiplier-quotient-register (S) from the arithmetical unit and also a so-called *indicator word*, the digits of which indicate the momentary state of a number of key points in the machine, such as the position of certain switches on the operator's desk, the momentary condition of punched-card in- and output equipment, and busy conditions of transfers. Though in PASCAL there is no automatic interrupt possibility, by interrogating the indicator

word—especially in administrative problems—the programmer is able to use his computing time very efficiently.

The single functional digit or address-interpretation digit (a.i.) for half the number of instructions (nos. 32–63) specifies whether the address part indicates a storage address or is an operand, thus saving selection time and a separate storage location in the case of operations with small positive integer numbers. For the other half of the set of instructions (except nos. 30 and 31) the presence of a 1 in this digit position changes the instruction into a so-called link-II instruction, which will be described later.

B. The Arithmetic Unit

Besides the registers A and S already mentioned, the arithmetic unit contains a third register M, the memory access register, through which all information to or from the memory passes (see block diagram, Fig. 2). A and M form the inputs for a diode-logic adder which at its outputs constantly holds the sum $(A)+(M)$ available.

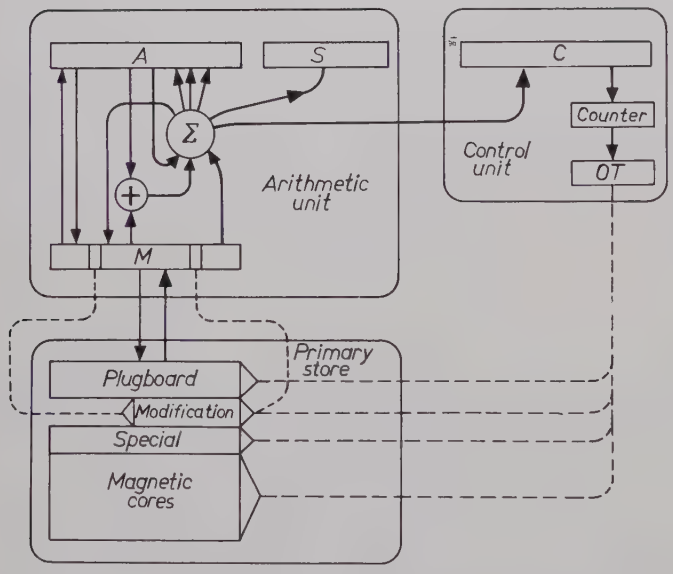


Fig. 2—Block diagram of PASCAL.

A large fraction of the traffic between the arithmetic registers passes through a multiple 6-way switch Σ, shown between the accumulator and adder. This switch may be entered from the adder, from A, and from M. Of the 6 sets of outputs, three sets lead to A, two of which are shifted over one digit position to the left or right for use during division and multiplication; the third set of connections brings the sum into A in the normal way. A fourth set of outputs of Σ leads to S, giving S the properties of a second accumulator; one set of outputs leads to the control register C enabling the use of the adder for modification of instructions. The remaining connections bring the output of Σ back into M, making it possible, *e.g.*, to use a memory address as an accumulator.

C. The Adder

The diode adder is constructed so as to have a 42-digit sum available at its outputs within 0.8 μsec after a change in the contents of the input registers A or M. The sign bit is already formed after 0.5 μsec. This enables the computer to complete one step of a restoring division within one basic period of $1\frac{1}{2}$ μsec, according to the scheme:

2 $[(A)-(M)]$ replaces (A) if signs of $(A)-(M)$ and (A) are equal,

2 (A) replaces (A) if signs of $(A)-(M)$ and (A) are different.

Thus, the sign bit of $(A)-(M)$ is to be formed in time for the control to choose between these two alternatives.

A basic multiplication step according to the scheme:

$\frac{1}{2}[(A)+(M)]$ replaces A if $S_{41} \neq S_0$,

$\frac{1}{2}\ (A)$ replaces A if $S_{41} = S_0$,

is then possible *a fortiori* within one basic period as the decision between the alternatives (depending on the least significant bit of S being equal to the sign digit or not) originates here from the preceding basic period.

In order to obtain the speed mentioned above with the basic circuits used in PASCAL a simple propagated carry could not be used as this would take up to $2\frac{1}{2}$ μsec. The adder, therefore, has been divided into blocks of 7 bits each, every block producing an output carry derived from its input carry and the carry-propagating or carry-generating conditions for the block. These were called "fast carries."

Thus, if a_{ij} and m_{ij} are the jth digits in the ith block of the registers A and M, then the carry-generating condition d_{ij} is expressed as[1]

$$d_{ij} = a_{ij} \cdot m_{ij}$$

whereas the carry-propagating condition e_{ij} may be written as

$$e_{ij} = a_{ij} + m_{ij}.$$

From these, the carry-generating condition for the ith block is found to be

$$D_i = \sum_{k=0}^{6} d_{ik} \prod_{l=0}^{k} e_{il}$$

and the carry-propagating condition,

$$E_i = \prod_{l=0}^{6} e_{il}.$$

[1] In the following formulas $\cdot$ and $+$ have been used for the logical operations $\wedge$ and $\vee$ (AND and OR respectively).

From these quantities the "fast output carry," *e.g.*, of block 3, Cb_3, follows as

$$Cb_3 = D_3 + Cb_4E_3,$$

where Cb_4 is the input carry of block 3 (see Fig. 3 for the arrangement of the blocks).

On a next-higher level, the carry-generating and carry-propagating conditions of a group of blocks are combined to produce "instantaneous carries" which are injected at four points into the adder.

The instantaneous carry CI_3 between blocks 3 and 2, *e.g.*, is found from

$$CI_3 = D_3 + D_4E_3 + D_5E_3E_4 + D_0E_3E_4E_5 + D_1E_3E_4E_5E_0 + D_2E_3E_4E_5E_0E_1.$$

For obtaining the sign digit early, similar measures are taken within block 0. This yields the required build-up speed for all digits of the adder with a sufficient safety margin.

This setup also allows a check on the functioning of the adder as carries arriving at the same point along different roads are compared and should be equal.

Afterwards, when fast transistors became available, it turned out that if transistors had been used in the carry-propagating circuits instead of tubes, the instantaneous carries could have been dispensed with and only two blocks forming fast carries would have been needed.

The adder is also used in a fast "comparison" instruction, in which the difference between the contents of A and a memory address n is tested without changing either (A) or (n). The result of this comparison is stored in two flip-flops[2] which may be interrogated later on in the program by one of three jump instructions reading:

jump if the last comparison gave $(A) > (n)$,
jump if the last comparison gave $(A) = (n)$,
jump if the last comparison gave $(A) < (n)$.

The contents of the 2 flip-flops are changed only when a new comparison instruction occurs.

Representative operation times may be found in Table I.

D. Modification of Instructions

When a new pair of instructions is extracted from the store, they go through M into A, while the contents of A temporarily are stored in the control register C. During the period the instructions are in M the selection for the modification registers (static *p-n-p-n* flip-flop registers) is set up. This selection circuit exists in duplicate as the situation may be different for the two word halves and modification of these halves is done at the same time, or is suppressed, as need may be. When the contents of the modification registers appear in M, the modified instruction pair appears at the outputs of the adder, then passes through Σ into C, in the same period transferring

[2] What actually is done, is: the sign bit of $(A)-(n)$ and of $(A)-(n)+p$ are stored; p being the least significant bit of the word.

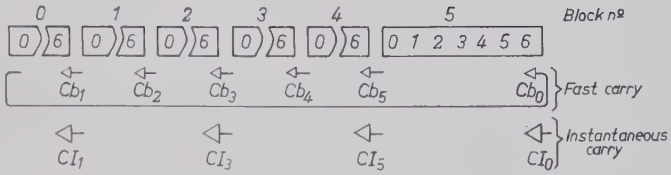


Fig. 3—Arrangement of the adder in blocks of digits, for obtaining so-called fast carries.

TABLE I

REPRESENTATIVE OPERATION TIMES,* INCLUDING MEMORY ACCESS FOR INSTRUCTION AND OPERAND

Operation	Fixed Point		Floating Point	
	Modified	Not Modified	Modified	Not Modified
+	$11\frac{1}{4}$ μsec	$9\frac{3}{4}$ μsec	$15\frac{3}{4}$–$65\frac{1}{4}$ μsec	$14\frac{1}{4}$–$63\frac{3}{4}$ μsec
−	$12\frac{3}{4}$ μsec	$11\frac{1}{4}$ μsec	$15\frac{3}{4}$–$65\frac{1}{4}$ μsec	$14\frac{1}{4}$–$63\frac{3}{4}$ μsec
×	$72\frac{3}{4}$ μsec	$71\frac{1}{4}$ μsec	$60\frac{3}{4}$ μsec	$59\frac{1}{4}$ μsec
÷	$74\frac{1}{4}$ μsec	$72\frac{3}{4}$ μsec	$62\frac{1}{4}$ μsec	$60\frac{3}{4}$ μsec

* As the two instructions per word share some of their micro-operations, these times need not be multiples of $1\frac{1}{2}$ μsec.

(C) back into A and putting into M what was to be written back into the cores in order to restore the instruction pair in memory.

Now, there is a difference in functioning between the even and the odd modification registers, which therefore could be termed index and increment registers, respectively. When an index register is used for modification, the instruction is written back into the core storage in the original, *i.e.*, unmodified form; when, however, an increment register has been used, the instruction is written back into storage in the modified form. One version is useful in the application of standard subroutines, the other one in counting or stepping-up procedures. The contents of the modification registers themselves are not altered in the process. As changing of the contents of modification registers occurs less frequently than reading them out, the somewhat slower switching speed of *p-n-p-n* transistors could be tolerated.

Parity-check bits, one per half word, which accompany every word in storage are either checked or generated when a word passes M. If the check fails, the machine will stop and indicate the reason.

E. The Control Part

The control part of PASCAL has as its main registers the 42-bit control register C, and the 11-bit address-selection register OT. Besides, there are 3 different counters, known as BT (4 bits), RT (3 bits) and ST (6 bits).

BT, the control counter, counts the 12 phases through which the control passes when carrying out one instruction pair. These 12 phases can be brought into 4 groups of three, which perform the following manipulations:

0, 1, 2. An instruction pair is extracted from the core-memory, modification selection is set up and modifications are extracted.

3, 4, 5. Modifications are performed and instructions written back in the correct way.
6, 7, 8. First operand extracted from the store and the operation executed by the arithmetic unit.
9, 10, 11. Second operand extracted from the store and operation executed.

It goes without saying that, because of the variety of instructions, many changes of this main pattern can be necessary, mostly leading to the skipping of certain phases of this procedure. As soon as the control starts the execution of a prescribed operation, the *RT* counter starts to work, controlling the various steps which the particular operation asks for while the *BT* counter remains in the phase it is in. If one of these steps needs to be repeated a number of times, this is governed by the *ST* counter (as in multiplications, shifts, etc.)

F. The Secondary Storage

Apart from the various forms of primary storage that have already been mentioned, PASCAL has access to a magnetic drum and magnetic tape as secondary storage. In problems like the solution of partial differential equations by iterative approximations, where both high speed and large storage capacity are required, it is important that the rate at which results can flow back and forth between the primary and the secondary store is not smaller than the processing rate in order that the computer can be kept busy all the time and that it is not hampered too much by the transfers taking place. Estimations of representative problems showed that a processing rate of more than 12 words/msec on the average (*i.e.*, approximately 10 short instructions in the iteration loop) would rarely occur in PASCAL.

Correspondingly a magnetic drum has been designed for 16384 words of storage together with a transfer unit which can effect transfers of 12.8 words/msec in both directions simultaneously between drum and cores. The drum revolution time is 10 msec and one drum track contains a block of 128 words. For convenience one may also think of the primary store as arranged in blocks of 128 words. Effectively, one drum track consists of two actual tracks on the drum, each containing 128 half words and treated in parallel by two read-write heads of which there are 256 in total, arranged in two opposite rows of 128. A buffer register T of 44 bits (42+2 parity bits) communicates serially (as two half-word shifting-registers) with the drum heads and in parallel with the core storage. In between half words on the drum tracks, gaps are left of 12-μsec duration, during which time a word may be put from the T register into the cores and a different word taken from the cores into the buffer. In the next period of 66 μsec the contents of the buffer are written on the drum surface while at the same time different heads read a new word serially into the buffer.

This transfer is activated by a transfer instruction specifying the respective block numbers. The transfer instruction does not wait for the zero address to pass under the heads, but starts with the first word available after the track selection has been effected and transfers the contents of one whole circumference, keeping track of the actual locations of the words. Only if the computer wants a number from the core store during one of the 12 μsec gaps is the computer blocked. In the very worst case, this blocking gives a time loss of 15 per cent.[3]

A similar arrangement exists for the use of magnetic tape. When using separate tape units for reading and writing simultaneously, the bidirectional information rate then is 2×8 instead of 2×12 words/msec on the average. One-inch tape is used carrying 16 tracks, which are twice 6 information tracks, 1 parity and 1 clock track.

G. Input and Output

As means for input and output the computer has the following facilities. As input means there are

1) A punched paper tape reader handling reels of 5- to 8-hole tape at a rate of 1200 characters per second. Reading is then done in blocks of arbitrary length after which about 2 or 3 character stopping distance is allowed for. By setting a switch the reader will work at 120 character/sec, then stop on the character. Conversion of floating addresses and of floating-point numbers by the input routine will just take place between the reading of two characters when reading at high speed. The tape is read at two reading stations; the information of the second reading station may be used for checking purposes.
2) Punched cards in Bull or IBM code. These can be read by a Bull Sorter D3, reading approximately 12 cards/sec, or by a Bull Tabulator *BS* at $2\frac{1}{2}$ cards/sec. In both cases there are two reading stations one of which may be used by the programmer for checking purposes. The punched-card reading stations all have their own magnetic-core buffer storage.
3) Magnetic tape. Up to 16 tape units (Ampex type *FR* 316) may be added to PASCAL.

As output means PASCAL has

1) A Teletype paper-tape punch, working at 50 character/sec.
1) An electric typewriter (IBM) printing 10 character/sec. This typewriter on the control desk will mainly be used for instruction of the operator.
3) The Bull Tabulator *BS* has in addition a card punch working at $1\frac{1}{4}$ cards/sec.
4) The Bull Tabulator *BS* is also a line printer printing $2\frac{1}{2}$ lines/sec of 92 characters each.
5) Magnetic tape.

[3] For programming reasons the computer is also blocked if it wants information out of a block that is being transferred.

6) For fast printed output an Anelex high-speed printer working off line from magnetic tape can be used.

II. The Instruction Code of PASCAL

With its 6 operation bits PASCAL has the possibility of 64 different instructions, 7 of which have not been used. Besides, the link-II facility provides the programmer with a further set of 30 interpreting instructions, the function of which the programmer can arbitrarily choose for his own purpose. A complete list of instructions is given in Appendix I; a number of more general remarks will be given in this section.

1) Due to the fact that there are two instructions per word PASCAL has a double set of jump instructions; even operation numbers jump to the 0_1, odd numbers to the 0_2 half of the specified address. Apart from the normal kind of jumps, 3 kinds of jump on the result of a comparison instruction already have been mentioned. Also there is an instruction that jumps if the overflow indicator flip-flop has been set. When the jump takes place the overflow indication is reset.

2) All input facilities have one operation number. The different varieties are specified by combinations of some of the address bits. The same holds for output.

3) Arithmetical operations exist in fixed-point and in floating-point versions. The floating-point version includes normalizing the results. Logical operations are bit-by-bit conjunction and disjunction.

4) Link-I instruction. This instruction facilitates calling in a subroutine in the normal way. If a link-I instruction specifies address n, then the following actions take place: A jump instruction to the instruction immediately following the link-I instruction is written in the 0_2 half of address n and the 0_1 instruction of address $n+1$ is carried out. In this way the subroutine is started. The subroutine then must end with a jump to the 0_2-position of its own first address. In the 0_1 half of address n, that information is copied which was in the same instruction pair together with the link-I, thus enabling the programmer to take extra information from the main program into the subroutine if wanted.

5) The function of the address-interpretation bit, which specifies whether the address part of the instruction is an address in the core memory or the operand itself, has already been mentioned; *e.g.*, in shift instructions the number of positions of the shift is found in the location specified by the address unless the a.i. bit is a 1. However, in some instructions this bit has a special meaning, *e.g.*,

a) With instructions that write something into memory an a.i. bit equal to 1 cannot be used; the machine would stop.

b) The count instruction. This instruction is normally a skip. Only with an a.i. bit equal to 1 does it become a jump to the instruction *following the next one*, in this way making possible a jump out of a loop. The idea is that by using an odd modification register with the count instruction its address part will be increased every time the instruction is passed in the program until overflow into the address-interpretation bit occurs.

c) The repeat instruction works more or less along the same lines. Normally it repeats the other instruction of the pair of instructions it is in. Only with an a.i. bit equal to 1 is it a skip and does the program go on. Also here, an odd modification register specified in the repeat instruction is added to the address part in every repetition until overflow occurs into the a.i. bit. A restriction to b) and c) is that these instructions must be in the 0_2 position of the pair.

d) For instructions 0–29 inclusive, the usual interpretation as shown in Appendix I applies only for a.i. equal to 0. These instructions, with an a.i. bit equal to 1 are so-called link-II instructions. Their action is as follows. They suppose a group of 30 words on fixed addresses in the store to have been set aside as a list of subroutines, every word of the list having in its 0_1 position a jump instruction to one of the subroutines whereas every subroutine is supposed to end in a jump to the 0_2 position of its own word in the list.

Now, if in the main program a link-II instruction occurs, *e.g.*, instruction 7 plus the a.i. bit, the control places in the 0_2 position of address no. 7 of the list a jump to that instruction of the main program immediately following the link-II and carries out the instruction in the 0_1 position of no. 7 of the list, *i.e.*, the jump to subroutine no. 7 of the list.

The address mentioned in the link-II instruction, *i.e.*, the address of the operand on which the subroutine is carried out, is copied in index register no. 6, where it can be found by the subroutine. In this way, the same group of instructions can be used by different programmers to write interpretive programs to their own needs, and solely by changing the list one can, within the same program, change from one set to another set of 30 subroutines.

6) Transfer instructions. As already mentioned, transfers in different directions can be done simultaneously. Those for communication between core and drum storages specify the respective block numbers. The transfer operations between core and tape storages take care of a number of varieties: single-block or double-block transfers, read tape forwards or backwards (this does not make any difference to the order in which, ultimately, the information arrives in the core storage), search for a block gap, proceed one blocklength (to avoid possible flaws on the tape after detection), erase, rewind, etc.

Some programming examples can be found in Appendix II.

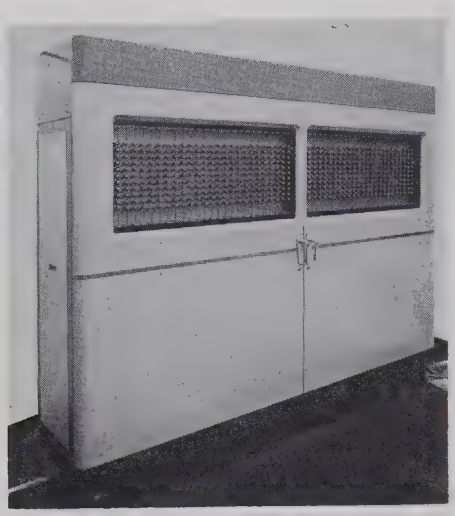


Fig. 4—Cabinet, housing PASCAL.

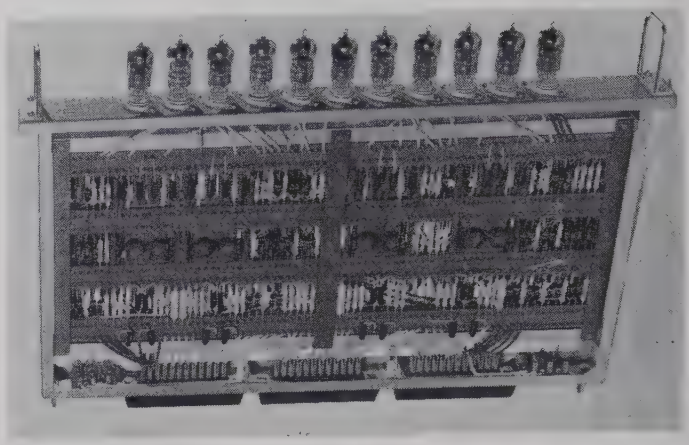


Fig. 6—One of PASCAL's pluggable units.

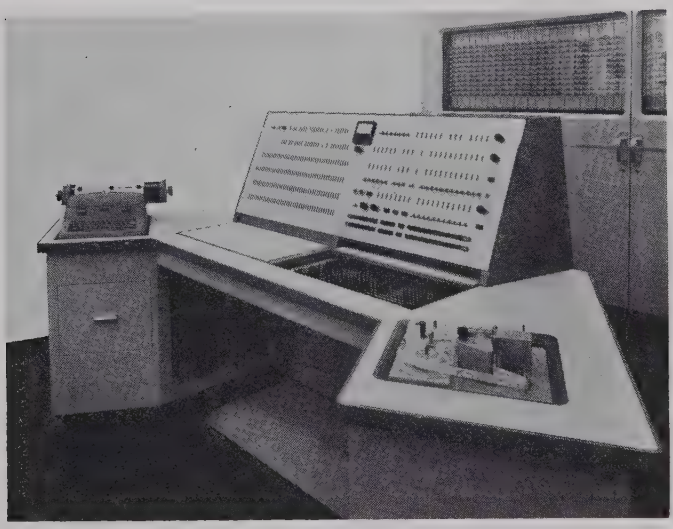


Fig. 5—Operator's desk for PASCAL.

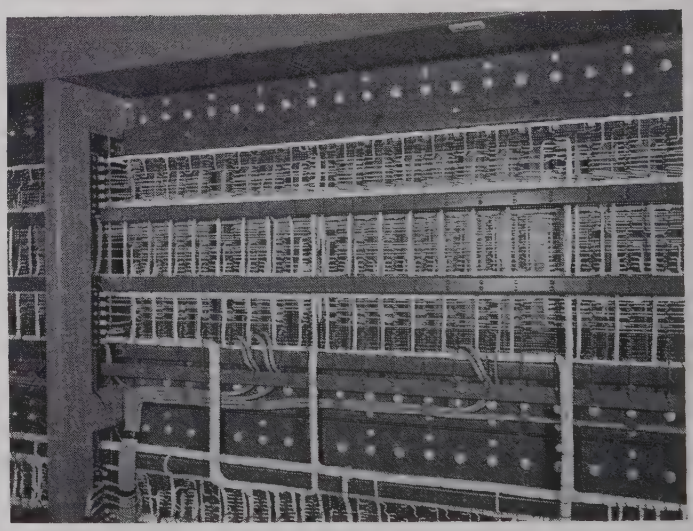


Fig. 7—Part of the wiring of PASCAL.

III. Constructional Details

PASCAL consists of a main unit, Fig. 4, housed in an airtight cabinet of $1.90 \times 2.70 \times 0.50$ meters, a drum cabinet of $0.90 \times 0.60 \times 1.20$ meters, and an operator's desk (Fig. 5).

As at the time the design was started, there were not sufficient high-speed transistors available, for some of the arithmetic circuits, tubes (Philips E 88 CC) have been used. (Also, tubes were used as cathode followers.) In all, the computer, not counting peripheral equipment, contains about 1200 tubes and about 8000 transistors (OC47.)

The total power consumption of this main unit is about 10 kw, and to cope with this amount of heat, a forced closed-loop air circulation system equipped with a watercooled radiator keeps the ambient temperature of the tubes below 50°C and that of the other circuitry below 30°C.

The computer is built as a number of pluggable units of a size that was adopted to give a convenient arrangement for the arithmetical section, where, in every register the circuitry that pertains to the same binary position has been taken together. (See Fig. 6.) In this way, this section contains 42 identical units and short leads and few contacts are ensured. Having pluggable units that are not too small was also considered an advantage with respect to error tracing in the machine.

The wiring for interconnecting these pluggable units, being surprisingly simple, could be brought outside the air-tight cabinet (Fig. 7), which makes most points of interest in the circuits accessible for measuring purposes at the back of the cabinet. Also, here the instruments and switches are found for performing marginal checks on the voltages which are stabilized to 1 per cent, most of them having, however, a tolerable margin of 10 per cent. The power unit has been so designed that even 100-volt pulses on the mains have no effect on the stabilized voltages.

A representative example of the type of circuitry that has been used everywhere in the machine is given in Figs. 8 and 9. Fig. 9 shows a three-stage counter, acting as a section of a larger counter and which will count

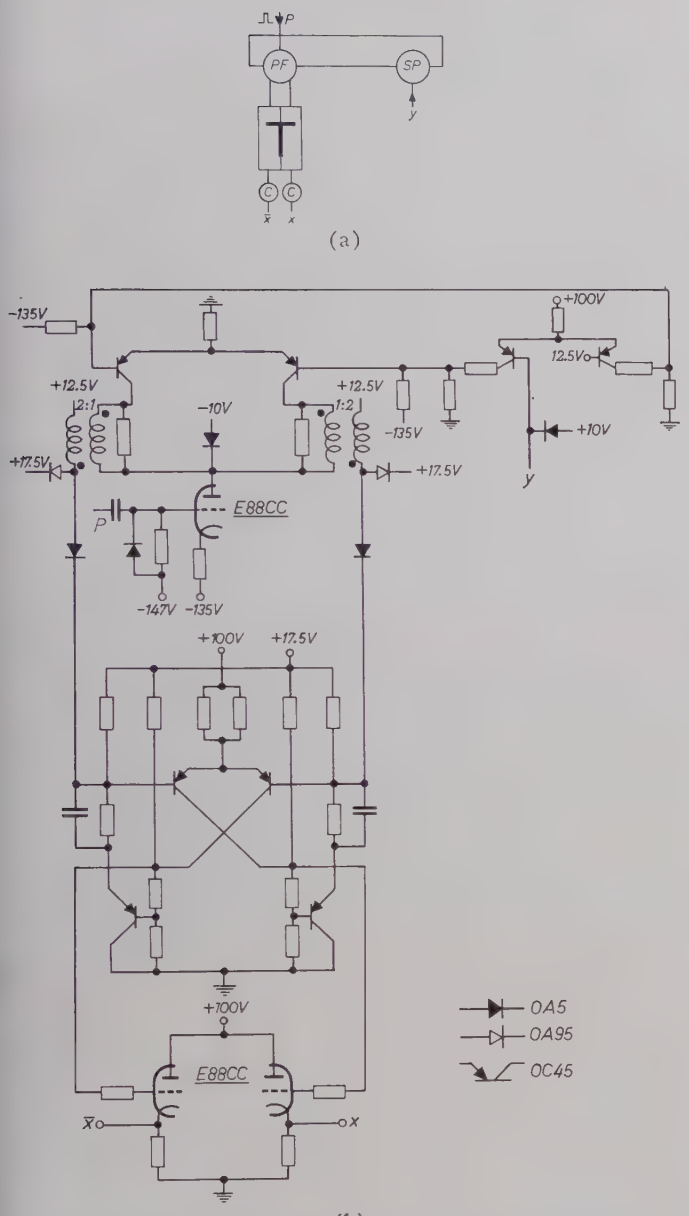


Fig. 8—Transistor flip-flop in PASCAL with associated circuitry.

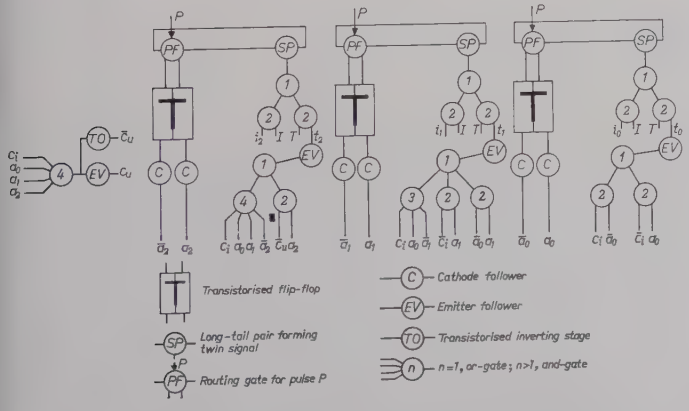


Fig. 9—Representative example of PASCAL's circuitry. A three-stage counter is shown that can be set into state $i_0 i_1 i_2$ under presence of a signal I and a clock pulse P and that, under presence of a signal T and an incoming carry c_i from the preceding stage, will count clock pulses P and deliver an outgoing carry c_u to following the stage.

clock pulses P as long as counting-command signal T and an incoming carry signal c_i from a preceding counter stage are present. The stages can be set into a prescribed state according to digits i_0, i_1, i_2 by applying a set-command signal I and a clock pulse P. It may be seen that every stage consists of AND-OR logic circuits receiving voltage levels as inputs where every two levels of logic are followed by emitter followers and a flip-flop with associated circuitry denoted by PF, SP and C. This latter part of the circuitry is shown in detail in Figs. 8(a) and 8(b). Fig. 8(b) shows in circuitry what is given as a block diagram in Fig. 8(a). The output voltage Y from the logic circuit (15 volts for a zero, 10 volts or less for a one), by means of a long-tail pair circuit of transistors, indicated as SP in Fig. 8(a), is formed into a twin signal corresponding to Y and $\overline{Y}$. This twin signal, by means of the routing gate PF directs the clock pulse P to either side of the flip-flop in order to make the output voltage X correspond to the input voltage Y. The outputs X and $\overline{X}$, delivered by cathode followers C, can be applied directly as inputs to other logic circuits. The internal delays in the whole circuit are just big enough that for clock pulses of 0.3 μsec duration, no difficulties occur when output signals of a particular flip-flop happen to be coupled back to its own input. Returning now to Fig. 9, it will be clear that, *e.g.*, the right-most counter stage is reset in the state it was in when the signal T is presented and c_i is not, whereas it changes its state on every clock pulse P when signals T and c_i are both present.

The operator's desk (Fig. 5) contains all switches and indicator lights necessary to run the machine. Mention was made already of the plugboard, address zero of which has special pushbuttons; the rest is beneath the desk surface and may be covered safely by a sliding lid. The contents of the three arithmetic registers are shown by small indicating tubes (DM71), the middle one A having an overflow bit extra. A fourth register (lowest left in the picture) continually displays the contents of an arbitrary memory location, the address of which can be set on switches.

Another set of switches may be used in error tracing: the computer may be set to stop on every instruction that agrees with the switches or on every address that agrees with (part of) the switches.

Also there are modes of operation in which the computer stops on every operation or on every phase of the control.

The instruction under execution is displayed as well as the situation transfers are in.

The meter in the middle of the picture indicates the average efficiency: it shows the percentage of time the computer does not spend in waiting (*e.g.*, for blocking periods during transfers, or for input- and output-procedures to end).

On the control desk is the paper-tape reader and an electric typewriter, and hidden in a drawer the paper-tape punch.

A separate cabinet, which measures 0.60×0.90×1.20 meters, houses the drum. The drum itself measures 0.40×0.25 meters; it makes 100 revs/sec. The surface is coated with iron oxide, and a row of 128 heads is placed on either side. When the drum is rotating these rows of heads are floating on an air cushion and follow the thermal expansions of the drum. A wind-vane mechanism ensures that when the drum stops the heads are removed to a safe distance.

The heads themselves are ferrite heads having a width of 1 mm. They carry a center-tapped coil of 100 turns and have an air gap of 30 μ. When in operation their distance to the drum surface is approximately 10 μ.

A special disk on the same shaft as the drum is engraved and delivers the clock track and a special zero sign. As this track delivers 330 kc pulses, this frequency has to be doubled for obtaining the clock pulses for the machine.

Appendix I

Instruction Code of PASCAL[4]

0, 1 Jump, absolute
2, 3 Jump, if $(A)>0$
4, 5 Jump, if $(A)<0$
6, 7 Jump, if $(S)>0$
8, 9 Jump, if $(S)<0$
10, 11 Jump, if last comparison gave >
12, 13 Jump, if last comparison gave =
14, 15 Jump, if last comparison gave <
16, 17 Jump, if overflow indication
Reset overflow indication

All even operation-numbers jump to the 0_1 part, all odd operation numbers to the 0_2 part of their addresses.

20 Output instruction
22 Transfer Cores→Drum
23 Transfer Drum→Cores
24 Input instruction
26 Transfer Cores→Tape
27 Transfer Tape→Cores
28 Interchange A and S
29 Skip
30 Count instruction
31 Repeat instruction

32 $(A)+(n)\rightarrow A$
33 $(A)-(n)\rightarrow A$
34 $(S)+(n)\rightarrow S$
35 $(S)-(n)\rightarrow S$
} fixed-point arithmetic operations

36 $(S)\times(n)\rightarrow A, S$
37 $(A, S)\div(n)\rightarrow S$
} A, S representing a double-length word

38 $(A)+(n)\rightarrow A$
39 $(A)-(n)\rightarrow A$
40 $(S)\times(n)\rightarrow A$
41 $(A)\div(n)\rightarrow S$
} floating-point operations

42 Shift (A) to the right
43 Shift (A) to the left
44 Shift (S) to the right
45 Shift (S) to the left
46 Shift (A, S) to the right
47 Shift (A, S) to the left

48 Logical AND, bit by bit
49 Compare (A) and (n), and store the possible results $>, =, <$
50 $(n)\rightarrow A$
51 $(n)\rightarrow S$

52 Stop
53 Logical OR, bit by bit

56 $(A)\rightarrow n$
57 $-(A)\rightarrow n$
58 $(S)\rightarrow n$
59 $-(S)\rightarrow n$

60 $(n)+(A)\rightarrow n$
61 $(n)-(A)\rightarrow n$
63 Link − I

[4] No attempt is made to give every detail.

Appendix II

Two Programming Examples for PASCAL

As a programming example, consider the computation of an unrounded scalar product X of two 10-vectors $\boldsymbol{P}$ and $\boldsymbol{Q}$.

$$\boldsymbol{P}\cdot\boldsymbol{Q}=\sum_{1}^{10} p_i q_i = X$$

The vectors are supposed to reside in addresses 100–109 and 110–119; the product is built up in 200 which is empty at the start.

Time	Address	Operation	Modification Register	a.i. bit	Operand Address	Remarks
84 μsec	α	→51	1*	0	99	p_i taken in S
		36	1	0	109	p_iq_i calculated
24 μsec	α+1	60	0	0	200	Σp_iq_i stored in 200
		30	1	0	2048–10	jump after 10 times
7½ μsec	α+2	— 0	0	0	α	
		52	0	0	0 ←	

* Modification register 0 always contains 0. Modification register 1 always contains 1.

During every cycle both operand addresses in α are increased by 1. The "count" instruction in $\alpha+1$ also increases its address by 1 but acts as a "skip" until overflow in its a.i. position occurs, *i.e.*, after 10 times, when it ignores the first instruction in $\alpha+2$ and steps to the second one. The time of one cycle would be 117 μsec.

As a second example, the use of a link-II as an interpretive instruction carrying out floating-point addition of complex numbers is given. The fixed addresses for the reference list of interpretive routines in PASCAL are the locations from 96 onwards and floating-point complex addition is assumed to be routine no. 7 of this list. The actual subroutine is supposed to be in locations from β onwards. Location $96+7=103$ in the reference list now contains the instruction $0 / 0 / 0 / \beta$ and supposing the complex number to be added is found in addresses 200 and 201, the main program (say instruction 0_1 in address α) just reads: $7 / 0 / 1 / 200$. This causes the control to carry out the 0_1 part of (103) and to complete this pair as

$$(103): \quad 0 / 0 / 0 / \beta$$
$$1 / 0 / 0 / \alpha$$

whereas the operand address "200" is copied in modification register 6.

The subroutine itself might use A and S as pseudo-accumulators for the real and imaginary parts. It finds the address of the real part of the operand in modification register 6. The subroutine might now read:

Time	Address	Operation	Modification Register	a.i. bit	Operand Address	Remarks
24	β	38	6	0	0	Add real parts
9 μsec		28	0	0	0	dump in S
24	β+1	38	6	0	1	Add imaginary parts
9 μsec		28	0	0	0	restore pseudo-accumulators
7½ μsec	β+2	1	0	0	103	Return via list to main program

The subroutine time (assuming 6 normalizing shifts per floating addition) would be 48+18 μsec, transfer times of the link to the subroutine and back 34½ μsec.

Esaki Diode NOT-OR Logic Circuits*

H. S. YOURKE,† MEMBER, IRE, S. A. BUTLER†, MEMBER, IRE, AND W. G. STROHM†, MEMBER, IRE

Summary—**A basic technique is presented which enables the development of Esaki diode NOT-OR logic circuits. Two embodiments of the basic scheme are discussed, which, when combined with an OR-DELAY circuit, provide a logically complete system. Emphasis is placed on the more economical of the two embodiments. A tolerance analysis is included, which demonstrates that the technique enables the practical design of logic circuits. The requirements placed on Esaki diode characteristics, and the speed limitations of the circuits, are discussed. Examples of working circuits are shown, including photographs of voltage wave shapes.**

* Received by the PGEC, February 17, 1961.
† Advanced Systems Div., IBM Corp., Yorktown Heights, N. Y.

I. Introduction

THE various Esaki diode logic circuits which have been proposed to date may be placed in two general categories. The first category makes use of the current-driven single Esaki diode latch circuit. The device is biased with a relatively high-impedance source (usually clocked) such as to exhibit two stable states and provide significant dc current to load circuits only when in the high-voltage state. The circuit will switch to the high-voltage state when the sum of the input currents and source current exceeds the peak cur-

rent of the Esaki diode. There have been several proposals[1,2] that this circuit be used to perform AND and OR current summation logic. Techniques of this type impose severe tolerance requirements on components, power supplies and clocked sources and realistic analysis demonstrates that only the OR function can be performed in a practical sense. The second category makes use of the twin Esaki diode voltage-driven latch circuit.[3] The circuit consists of two series Esaki diodes driven at both ends of the series combination by two clock pulses of opposite polarity such as to guarantee that one, and only one, of the diodes assumes the high-voltage state. The input-output node connecting the two diodes will be positive or negative and the state assumed is determined by seeding from preceding stages. Majority logic may be performed on the summation of the inputs. As in the case for the circuits of the first category, the theoretical limit of current available to drive load circuits approaches the difference between the peak-to-valley currents of the Esaki diodes used, and, in practice, is limited to a small fraction of that. However, drive requirements can be made small, but only at the expense of extremely tight tolerances on the Esaki diodes (including capacitance and dc characteristics in the high- and low-voltage regions). This technique also imposes severe requirements on the regulation of the clocked voltage sources.

The technique proposed here falls into the first category, but operates in a new mode and has the following advantages:

1) The technique results in a logic inverter which can perform the NOT-OR function. Combined with the use of an OR-DELAY block, a logically complete system is developed.
2) The ability to drive load circuits is not limited by the peak-to-valley current ratio of the Esaki diode.
3) The tolerance requirements on the Esaki diodes are only stringent with regard to the peak current value. Tolerances can be relatively loose on all other aspects of the Esaki diode characteristics.
4) The technique does not impose unrealistic tolerance requirements on clock pulse amplitudes or wave shapes, and over-all component tolerances are reasonable.

II. The Basic Technique

A circuit illustrating the basic technique is shown in Fig. 1. Also shown in Fig. 1 is a V-I characteristic for the Esaki diode and associated load lines. The Esaki diode is biased to an essentially constant current I_s by the use of the dc supply $+V$ and the relatively large resistor R_s. In the absence of input current or clock current, the circuit exhibits the two stable states 1 and 2.

If, prior to the application of the clock pulse, no input pulse has occurred, the circuit will be in state 1. During the clock pulse time, the circuit will switch to state 3 and after the clock pulse will go to state 2. During the transition from state 1 to state 3, a charge will be delivered to capacitor C and the load (R_L, C_L) of magnitude $C(V_3 - V_1)$. The peak current available to provide this charge can approach (assuming small diode capacitance) $I_s + I_{\text{clock}}$ minus the diode valley current.

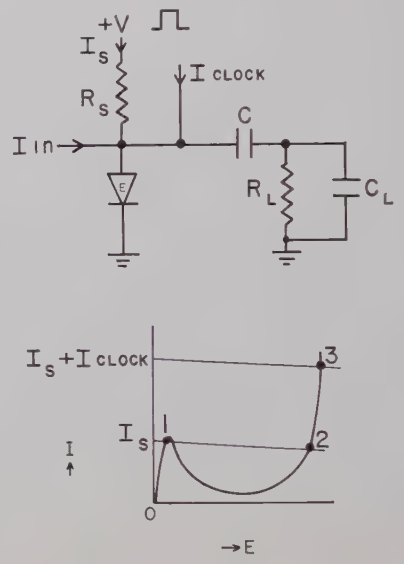


Fig. 1—Circuit illustrating the basic technique.

If, prior to the application of the clock pulse, an input pulse has occurred, the circuit will be in state 2. Then, during the pulse, the transition will be from state 2 to 3. During this transition, only a small charge of magnitude $(V_3 - V_2)C$ will be delivered to capacitor C and the load (R_L, C_L).

The circuit, therefore, performs as a logic inverter providing a significant pulse output only when there has been no pulse input. The NOT-OR function is performed by using more than one pulse input. Any logic embodiment must, of course, provide means for resetting the diode.

III. A Potentially Economical Embodiment

A variety of embodiments of the basic scheme are possible in the development of logic circuits. One of the more attractive of these is shown in Fig. 2. The associated three phase current clock waveforms and relative timing are also shown in Fig. 2.

Consider a circuit in the high-voltage state to be in the 0 state (will read out 0) and a circuit in the low-voltage state to be in the 1 state (will read out 1). If a circuit is in the 1 state when a positive clock pulse is applied, it will be set to the 0 state and will deliver a

[1] G. W. Neff, S. A. Butler, and D. L. Critchlow, "Esaki (tunnel) diode logic circuits," *Digest of Tech. Papers, 1960 Solid State Circuits Conf.*, pp. 16–17.
[2] M. H. Lewin, A. G. Samusenks, and A. W. Lo, "The tunnel diode as a logic element," *Digest of Tech. Papers, 1960 Solid State Circuits Conf.*, pp. 10–11.
[3] E. Goto, *et al.*, "Esaki diode high-speed logical circuits," IRE Transactions on Electronic Computers, vol. EC-9, pp. 25–29; March, 1960.

pulse to the stages in the next succeeding phase setting them to the 0 state. A negative clock pulse will reset the circuit. The negative pulses coupled to adjacent circuits during reset will not affect the state of these circuits (this is discussed in Section IV).

During the positive clock pulse time, a circuit in the 1 state will deliver positive pulses to preceding as well as succeeding stages. By virtue of the clock pulse sequence, the preceding stages are already in the 0 state and will not be affected. The sequence also assures that a positive pulse will not effectively propagate beyond stages in the next succeeding phase.

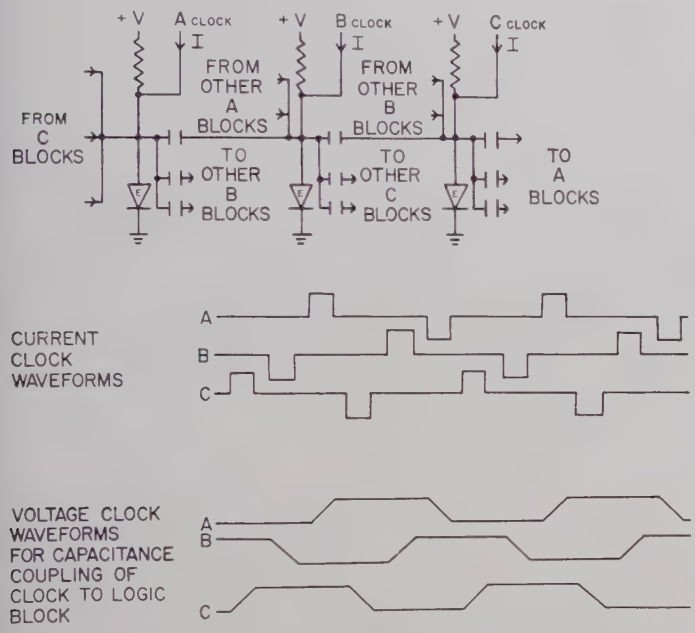


Fig. 2—Circuit and clocked source waveform illustrating a potentially economical technique for performing NOR logic.

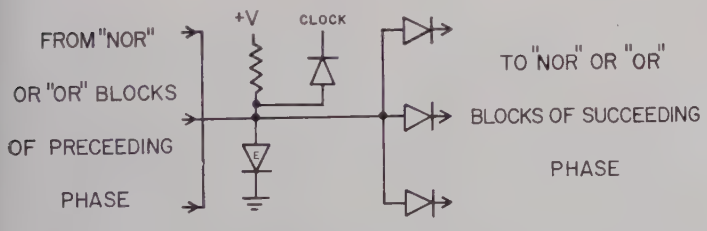


Fig. 3—A practical OR-DELAY circuit for use with the NOR circuit of Fig. 2.

The circuits described above perform the NOT-OR logic function; and if there were no restrictions on their interconnection, would provide a logically complete system. However, because of the required clock sequence, blocks of one phase can only be connected to blocks of the next succeeding phase. A simple delay circuit would make the system logically complete.

Several OR circuit techniques have previously been described in the literature which can provide the OR-DELAY function in this system.[1,2] An example of a practical (designable) OR-DELAY block for use in this system is shown in Fig. 3. This block is dc coupled to succeeding stages. Although other coupling techniques are possible, the block is connected to succeeding stages via normal rectifying diodes. This is done in order to minimize the tolerance requirements on the OR circuit and the circuits connected to it. A rectifying diode is shown in series with the clock line to illustrate the fact that the positive (set) portion of the clock source is decoupled from the circuit and only the reset portion is used.

The operation of the circuit is best described by the following example. Assume that the OR circuit of Fig. 3 is in the *B* phase (refer to Fig. 2) and has inputs from NOR blocks in the *A* phase and outputs to NOR blocks in the *C* phase. Just prior to the *A* set pulse, the *B* OR block has been reset to the low-voltage state and the *C* NOR blocks are in the high-voltage state (coupling diodes are back biased). During the *A* set pulse, the OR block may receive a 1 input pulse which will set it to the high-voltage state (coupling diodes near or at zero bias). During the *C* reset pulse, the coupling diodes become forward biased and deliver current to the *C* NOR blocks. The *C* blocks will then return to the high-voltage state after the *C* reset pulse. It should be noted that, although in general, the *C* reset pulse will be sufficient to override the input current from *B* OR blocks, this is not a requirement. The *B* set pulse is decoupled from the OR block in order to prevent the reading out of a 1 when a 1 input has not occurred. The logical sense of the dc level for the OR circuit is opposite to that of the NOR circuit. The OR circuit is in the 1 state (reads out at 1) when it is in the high-voltage state.

For purposes of minimizing the power dissipation and the requirements on the absolute levels of clock source distribution lines, it is desirable that the clock current be coupled to each circuit via relatively small capacitances driven by the voltage waveforms shown in Fig. 2. For the experimental work reported on here, the clock current was formed via resistor coupling to voltage clock lines having the same waveforms as the current clock waveforms in Fig. 2. This was done in order to permit maximum flexibility of the current amplitude and wave-shape for purposes of experimental analysis.

IV. Tolerance Analysis

In order to obtain a realistic evaluation of a circuit technique, the tolerance requirements must be known. In the analysis made here, the tolerances on all components are taken to their worst case simultaneously; worst case circuit configurations are also considered. The approximations and generalizations made are conservative. The component tolerances assumed are stringent but not unrealistic. It is likely that a more refined statistical analysis would result in significantly reduced tolerance requirements.

The analysis is made using a typical germanium Esaki diode *V-I* characteristic. The capacitors are considered to have tolerances of ± 10 per cent. The more demanding tolerances of ± 4 per cent on each are assigned

to the dc current supply and the peak current value of the Esaki diode. A typical *V-I* characteristic for a germanium Esaki diode is shown in Fig. 4 (the plot is a measured characteristic of a 1-ma diode). The load line formed by the high impedance source is shown as a band to represent the tolerance spread of the dc current supply and the peak current value of the Esaki diode. Fig. 5 shows an extended *V-I* characteristic of the same diode.

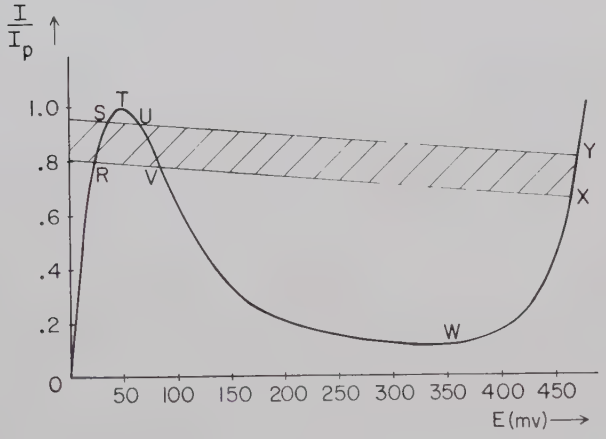


Fig. 4—Typical *V-I* characteristic for a germanium Esaki diode with a band illustrating the tolerance spread of the dc load line.

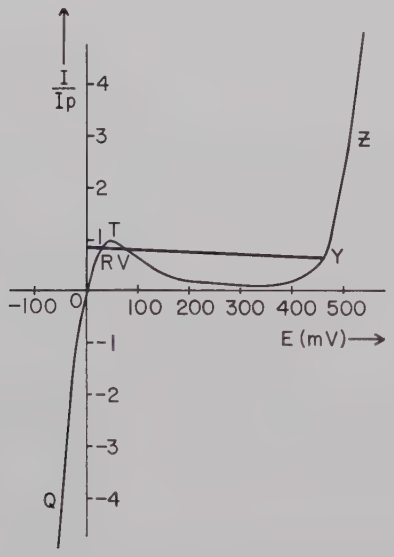


Fig. 5—Extended *V-I* characteristic for the diode of Fig. 4.

There are three cases to be considered. The reliable transfer of a 1 which establishes the lower limit on the load line. The reliable transfer of a 0 which establishes the upper limit on the load line. The third requirement is that the reset pulse does not propogate.

A. Reliable Transfer of a 1

This is the case where the most significant tolerance requirements exist. The term "branching factor" n is used to designate the maximum number of outputs a circuit may have; it will be assumed in this analysis that the maximum number of inputs is equal to the maximum number of outputs. The most heavily loaded case for a circuit having a branching factor of n is shown in Fig. 6(a). Diode D_1, when driven from its low- to its high-voltage state must, in turn, drive the n diodes on the right. The n diodes on the right are shown together because they require about the same voltage to assure their switching. If one of these diodes switches before the others, it essentially decouples itself and makes more drive available for the others. The capacitor n^2C represents the sum of the output capacitors of the n diodes. It is shown grounded because the capacitors it represents are connected to diodes that do not switch or change voltage at this time. In order to accomplish switching, the voltage across the n diodes must be increased beyond the intersection of the load line and the negative resistance portion of the *V-I* characteristic (this assumes negligible inductance). Point *V* (Fig. 4) is the critical intersection point in this case. There could be additional inputs to the diode nD coming from circuits that are already in their high-voltage state. The capacitors connected to these circuits offer no additional loading because the voltage at these circuits increases at least the same amount as that required to switch the diodes nD.

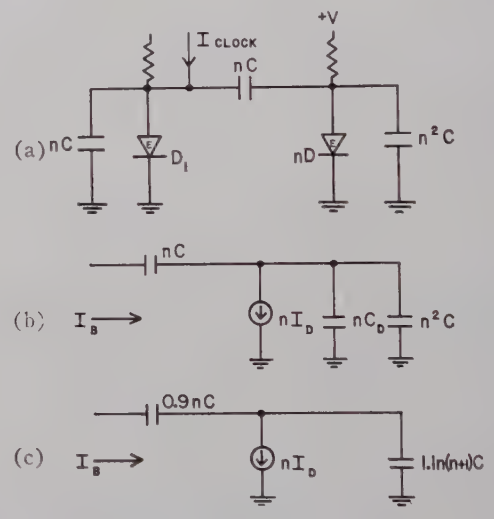


Fig. 6—Equivalent circuits for branching factor of n.

An equivalent circuit of the right-hand portion of the circuit of Fig. 6(a) is shown in Fig. 6(b). The capacitor nC_D represents the diode capacitance between points *R* and *V*. I_B is that portion of the drive current that flows through the load circuit while D_1 is being set by the clock pulse. The current nI_D represents the portion of I_B that is shunted through the conductance of the diode nD (it is the current difference between points *R* and *T* on the *V-I* plot in Fig. 4). It should be noted that I_D is the largest current difference that exists during the time that the operating point moves from

point R through point V and is therefore a conservatively chosen value.

In the worst case, the voltage across the n diodes must increase from point R through point V by the time the voltage across diode D_1 increases from point S to point Y. V_Y is used instead of V_Z (see Fig. 5) because the diode junction capacitance increases rapidly and the rate of voltage change decreases as the voltage increases beyond point Y. This means that the ratio of the voltage increase across the capacitors n^2C+nC_D to the increase across nC must be at least $V_{RV}/(V_{SY}-V_{RV})$. Assuming that the coupling capacitors are of the same size as the diode capacitances and can vary ± 10 per cent from some nominal value, the worst case can be represented as shown in Fig. 6(c). Succeeding calculations demonstrate that the lower practical limit on the magnitude of the coupling capacitors is close to the diode capacitance.

A value of I_B can be determined that will satisfy the requirements for the voltage ratio $V_{RV}/(V_{SY}-V_{RV})$ described above. With I_B constant, the voltage across $0.9\ nC$ at any time T while it is charging is

$$V_{0.9nC} = \frac{I_B T}{0.9nC}.$$

The voltage across $1.1n(n+1)C$ is

$$V_{1.1n(n+1)C} = \frac{(I_B - nI_D)T}{1.1n(n+1)C}.$$

The ratio of these two voltages is

$$\frac{V_{0.9nC}}{V_{1.1n(n+1)C}} = \frac{V_{SY}-V_{RV}}{V_{RV}} = \frac{I_B}{I_B+nI_D}\cdot\frac{1.1}{0.9}(n+1).$$

I_B from this equation is

$$I_B = \frac{nI_D}{1-\dfrac{V_{RV}}{(V_{SY}-V_{RV})}\cdot\dfrac{1.1}{0.9}(n+1)}.$$

The maximum theoretical branching factor under these worst case tolerance conditions occurs if I_B is infinite. I_B approaches infinity as the denominator of the fraction approaches zero; thus,

$$n = \frac{V_{SY}-V_{RV}}{V_{RV}}\cdot\frac{0.9}{1.1} - 1.$$

Using the V-I plot shown in Fig. 4, n would be

$$n = \frac{440-60}{60}\cdot\frac{0.9}{1.1} - 1 = 4.2.$$

A more realistic value of n would be 3. I_B under these conditions would be

$$I_B = \frac{3(0.2I_{peak})}{1-\dfrac{60}{440-60}\dfrac{1.1}{0.9}(4)} = 2.6I_p.$$

The capacitor nC and the capacitance of the diode D_1 [Fig. 6(a)] must be charged to a voltage of about V_Y. Calculations show that this charge is about 1.5 times that supplied by I_B. The total drive then would be

$$I_{drive} = 2.5I_B = 6.5I_p.$$

This again is conservative because the current difference between the load line and the V-I plot between points V and X was neglected. It would essentially add to the drive current during the time that the diode D_1 is switching.

B. Reliable Transfer of a 0

The second case to be considered is that if n diodes, already in their high-voltage state, are driven farther into their high-voltage state, they must not switch one diode from its low- to its high-voltage state. This case is shown in Fig. 7. It is assumed that all circuits have n output capacitors and that they are grounded if they are not used. Less than half of the voltage increase developed across the n diodes appears across the single diode. This is due to the voltage dividing action of the two capacitors nC on the right and the fact that some of the current that flows through the coupling capacitor is shunted by the conductance of the diode. This establishes the upper limit on the load line as shown in Fig. 4 (*i.e.*, V_{SU} must be at least $\frac{1}{2}$ the voltage V_{XZ}).

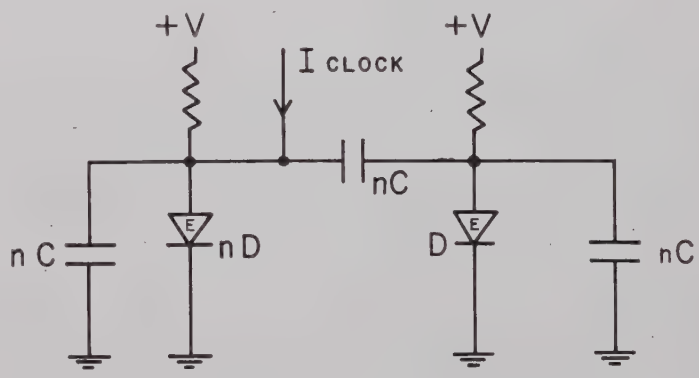


Fig. 7—Equivalent circuit for reset analysis.

C. Reset Must Not Propagate

The third and last case considered is when n diodes resetting from their high- to their low-voltage state must not reset a single diode. The worst condition for this case is also shown in Fig. 7. The voltage decrease that appears across the n diodes on the left is about V_{QY}. A voltage decrease of V_{YY} is necessary to reset the single diode from its high- to its low-voltage state. Since V_{QY} is considerably less than 2 V_{YY}, this case presents no problems.

V. Tolerance Requirements on the Esaki Diode

In the above tolerance analysis, a specific tolerance value was assigned to the Esaki diode only with regard to the peak current magnitude (± 4 per cent). At present in our own laboratories, diodes are being made with a tolerance of ± 1 per cent on peak current. Other tolerance requirements implied by the previous analysis are as follows: 1) A requirement is placed on the minimum magnitude of the voltage in the high-voltage region (point Y, Fig. 4). There is no limit on the maximum. 2) An upper limit is placed on the dynamic impedance in the high-voltage region (beyond point Y, Fig. 4). The nonlinear behavior of the junction capacitance contributes to the reduction of the dynamic impedance in this region. 3) A minimum slope in the negative resistance region, just to the right of the peak current point, is required. There is no maximum requirement. 4) A maximum limit is placed on the Esaki diode junction capacitance in the low-voltage region (this is a speed limitation only, the circuits can be designed for any magnitude of junction capacitance). There is no minimum requirement.

The analysis does not specifically imply any requirement on the peak-to-valley ratio of the diode other than that necessary to produce bi-stability. It is recognized, however, that the other requirements are related to some degree to the peak-to-valley ratio. Also, the ratio partially determines the minimum propagation delay.

It has been our experience that the above-mentioned requirements on the Esaki diode characteristics are not severe.

VI. Speed Limitations

The propagation delays (clock pulse widths) for the type of circuits described above are theoretically limited by:

1) the time required for a circuit to self-set after receiving inputs (as distinguished from clock pulse set).
2) the time required for a circuit to be reset by the clocked source.

The reset time is governed by the reset clock pulse amplitude and can be made arbitrarily fast. However, assuming the set and reset pulses have the same period, a rough approximation to the minimum propagation delay, T, is twice the self-set time.

An approximate formula for the self-set time is:

$$T = \frac{VC_D(N+1)}{SI_p} \cdot \frac{R}{R-1},$$

where

V = the voltage difference between the threshold point (Point V, Fig. 5) and the high-voltage region.
I_p = the peak current of the Esaki diode.
R = the ratio of I_v (Point V, Fig. 5) to the valley current of the diode.
C_D = the capacitance of the diode.
N = the sum of the inputs and outputs.
S = the ratio of I_v to I_p.

The formula assumes that the coupling capacitors are equal to the diode capacitance.

Assuming $N=6$ and a germanium Esaki diode where $V=0.4$, $S=0.75$, and $R=5$ the minimum propagation delay would be

$$2T = 9.3 \frac{C_D}{I_p}.$$

For a low capacitance diode of 1 pf/ma, the propagation delay could approach 9.3 mμsec.

Other embodiments of the basic scheme provide smaller theoretical limits on the propagation delay. However, the practical problems associated with packaging and the distribution of clocked sources will unquestionably impose more significant speed limitations than those of the circuit.

VII. Examples of Working Circuits

A. Three-Stage Closed Loop

In Fig. 8 a circuit is shown which consists of one block (A phase) driving three blocks (B phase), each of which drives the same block (C phase). The output of the C phase block is tied to the previous A phase block to form a closed loop. The circuit was designed to emphasize the tolerance problems discussed in Case I and Case II of the tolerance analysis. Every block has three output capacitors which were grounded when not used for logical connectives. Fig. 9 shows the wave shapes observed for this arrangement. The three clock phases are shown in the upper three waveforms followed by the waveforms observed at the points indicated. The blocks [Fig. 9(a)–(c)] alternately propagate a 1 and 0 for each successive clock cycle, as can be seen in Fig. 9. The diodes used were 1-ma peak current germanium Esaki diodes having an average capacitance of about 30 pf. The maximum reliable propagation delay (separation of adjacent clock pulses) was approximately 150 mμsec as discussed in Section VI.

B. Half Adder

A block diagram for a half adder is shown in Fig. 10. The logic for the NOR and OR blocks have been explained previously. The waveforms shown in Fig. 11(a)–(c) show clock phase C, inputs X and Y, and outputs SUM and CARRY for various input conditions. Note that the CARRY output is from an OR block and

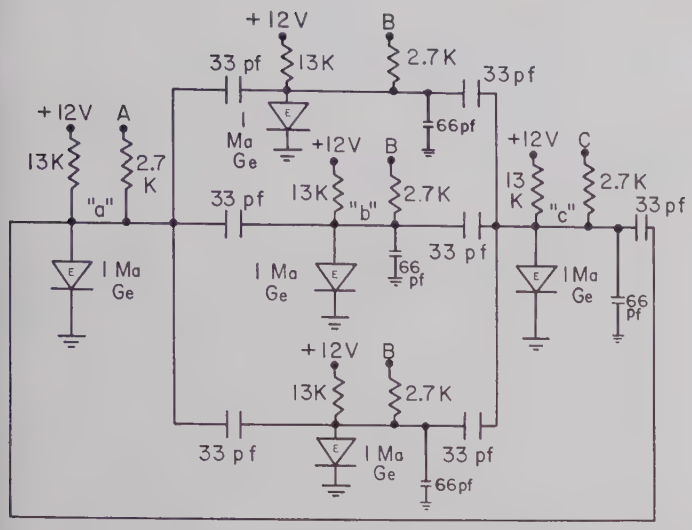


Fig. 8—Circuit diagram of a three stage closed loop.

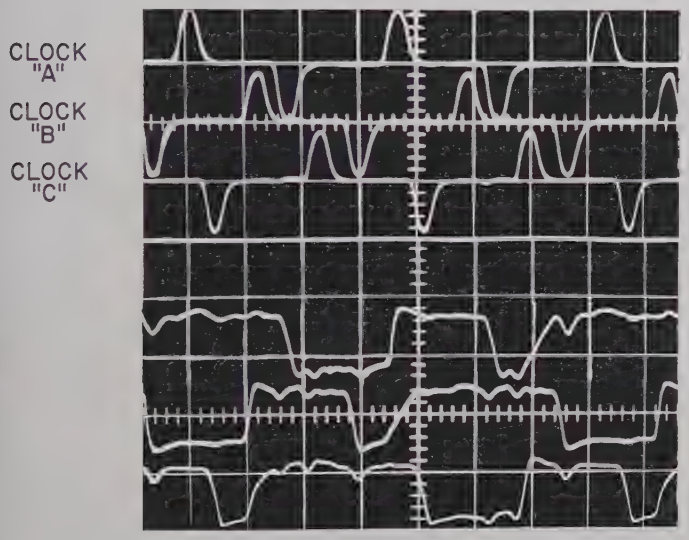


Fig. 9—Waveforms for circuit shown in Fig. 8. 0.2 μsec/cm; clocks 10 v/cm; others 0.5 v/cm.

the SUM output is from a NOR block, so that the logical sense of the levels is opposite. Inputs X and Y are supplied by NOR blocks.

VIII. Another Embodiment

Another configuration for an Esaki diode NOR block is shown in Fig. 12. Although this circuit is obviously more expensive than the previous block due to the diode coupling, it offers advantages in speed of operation and in tolerances. This block was investigated experimentally and was found to be quite tolerant of component and power supply variations. By the use of gallium arsenide Esaki diodes and conventional germanium coupling diodes, a branching factor of three is easily obtained.

Consider the two phase clock system shown in Fig. 13(a). When a stage is receiving an input from a preceding stage, the succeeding blocks are either in, or being

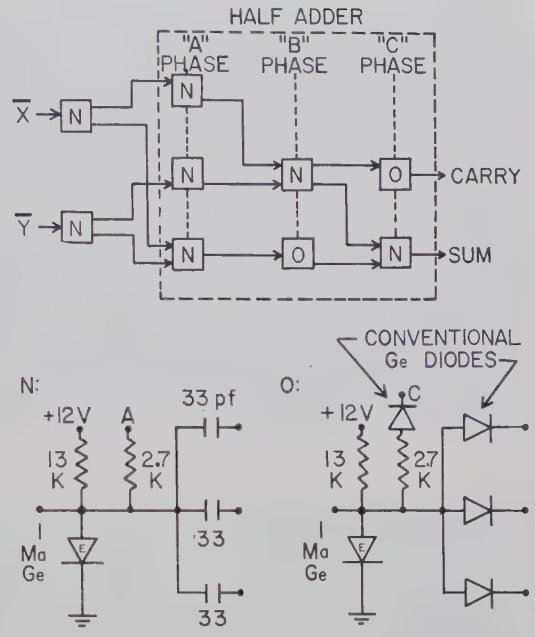


Fig. 10—Block diagram of half adder.

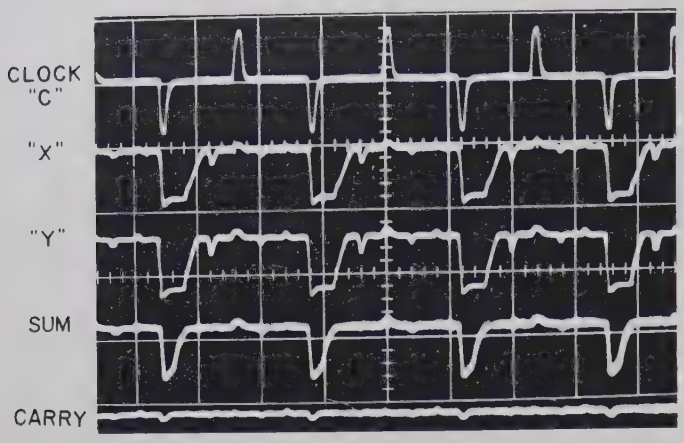
driven to, their high-voltage state. The coupling capacitors are essentially isolated during switching due to the reverse biased coupling diodes. Therefore, the driven diode is loaded only by the relatively small Esaki diode capacitance C_d (assuming low capacitance coupling diodes). This reduces the input drive required. Also since the self-set time is limited only by the diode capacitance, the theoretical limit on speed of operation for this arrangement, assuming a total of six inputs and outputs, is about six times greater than for the previous arrangement shown in Fig. 2.

The problem of rejecting small pulse inputs (discussed previously in Section IV-B) is also improved by this arrangement. Due to the nonlinear characteristics of the coupling diodes, small pulses that occur can be almost completely rejected.

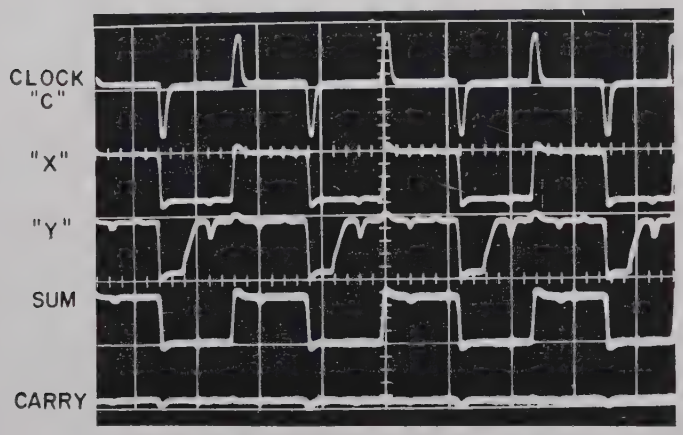
Since the coupling capacitors are isolated when a stage receives an input, they are not charged during the self-set time. They receive their charge when the succeeding stage is reset, and are discharged by their own reset clock pulse.

A three phase clock system could be used for this arrangement in order to avoid the delay necessary for a circuit to reset. This clock arrangement is shown in Fig. 13(b). However, even though the reset time is no longer "dead time," this technique does not enable significant reduction in self-set time and the two phase system may be faster. A resistor could be used in place of diode D-1, but more clock drive would be required to guarantee switching.

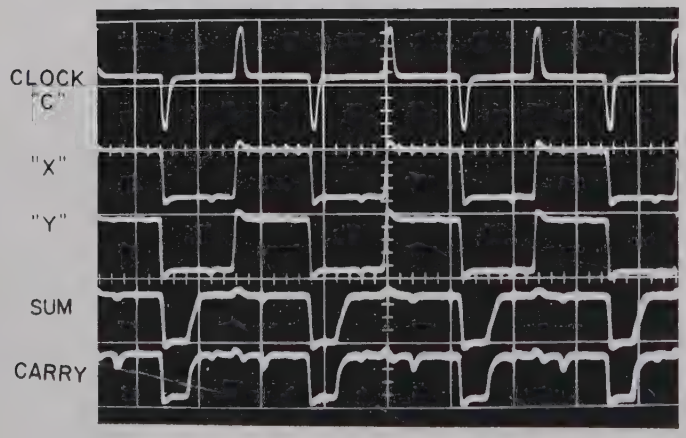
Other embodiments of the basic scheme are possible. However, the two circuits discussed appear to represent the extremes of economy and high speed potential.



(a)



(b)



(c)

Fig. 11—(a) Waveforms for half adder circuit shown in Fig. 10. 0.5 μsec/cm; clock 10; v/cm; others 0.5 v/cm. $X=0$ and $Y=0$. (b) $X=1$ and $Y=0$. (c) $X=1$ and $Y=1$.

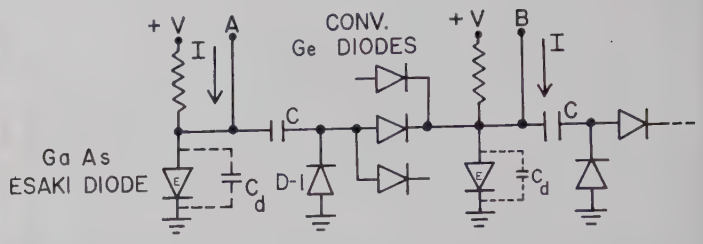
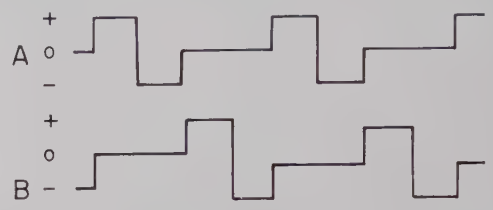
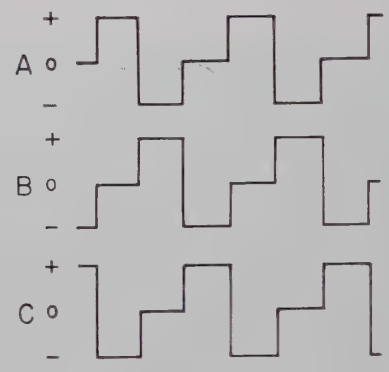


Fig. 12—Another configuration for an Esaki diode NOR block.



(a)



(b)

Fig. 13—Clock arrangements for circuit shown in Fig. 12.

IX. Conclusion

The circuit techniques discussed in this paper provide a means for the practical design of computer circuits using Esaki diodes. The significance of these techniques, as well as other Esaki diode logic techniques, is not at all obvious at this time. Their ultimate value will depend on how they compare with other methods of constructing computer circuits; especially with regard to the trend towards integrated circuits and devices.

Acknowledgment

The authors acknowledge the discussions with Dr. D. L. Critchlow and R. J. Sahulka and G. E. Simaitis which were helpful in preparing this paper. The assistance of Miss A. S. Zakarias and K. K. Smalley and R. Taylor is also acknowledged.

Logic Circuits Using Square-Loop Magnetic Devices: A Survey*

JOHN L. HAYNES†

Summary—The past decade has been a productive period for development in the field of large-scale digital computers. Magnetics has played an increasingly important part in these developments [1]–[8]. Magnetic cores, tapes, and drums have found almost universal acceptance for memory functions. Some future computers will undoubtedly use magnetic logic and control circuits. This survey is a capsule view of twenty-four square-loop magnetic logic circuits which have been proposed or developed so far, with a brief description of the way each circuit or circuit family meets the requirements of logic circuitry. All circuits are treated with a consistent terminology, and the generic relationships among circuits are stressed. Included in this survey are parallel and series transfer core-diode schemes, core-transistor schemes, and all-magnetic schemes of various topologies.

I. Introduction

IT IS the intention of this paper to present some techniques which have been suggested for using magnetic circuits for digital computer logic and control functions. Section II is devoted to a brief introductory discussion of magnetic circuit relationships and the terminology used in the survey. Those familiar with digital magnetic circuitry can go directly to Section III, which contains a summary of some twenty-four square-loop magnetic logic circuits and their generic relationships.

II. Terminology and General Comments on Digital Magnetic Circuits

The magnetic structures used for the square-loop logic schemes to be discussed either are sintered ceramic ferrites, or are built up of thin ferromagnetic materials. Both classes of material possess a flux vs magnetomotive force characteristic, as shown in Fig. 1.

If this hysteresis loop is traversed, starting at $\phi = -\phi_r$, $F = 0$, three properties of note are encountered. First, as F is increased, very little change in ϕ occurs unless F is larger than F_1. That is, there is a threshold of minimum coercive force. Second, continued application of a coercive force of F_2 or greater causes the flux to change to a magnitude of ϕ_s. (The greater the excess field, $F - F_2$, the faster ϕ will switch from $-\phi_r$ to $+\phi_s$.) At this point, very little change in flux beyond ϕ_s is possible. The material is saturated. Third, if F is now reduced to zero, the flux falls back from ϕ_s to a remanent state ϕ_r. (The difference between ϕ_r and ϕ_s is termed elastic or reversible flux.)

* Received by the PGEC, February 29, 1960; revised manuscript received, January 19, 1961. The work reported in this paper was performed as a part of Contract AF 33(616)-6303 entitled "Philosophy of Logic Realization," sponsored by Wright Air Dev. Div.

† Precision Instrument Co., San Carlos, Calif. Formerly with Stanford Res. Inst., Menlo Park, Calif.

The ϕ, F properties just described are observed as the terminal voltage-current characteristics of a network linking a ferrite or ferromagnetic metal structure. The most frequently used structure in magnetic logic circuits is a "toroid" of essentially rectangular section (see Fig. 2). Neglecting parasitic reactances, the relationships in the ith winding are:

$$E_i = N\frac{d\phi}{idt}$$

$$F = N_i I_i.$$

E_i and I_i are related through the external circuit parameters, F and ϕ through the ϕ-F curve for the toroid.

To standardize the discussion of circuits, the following terminology is used (see Fig. 3).

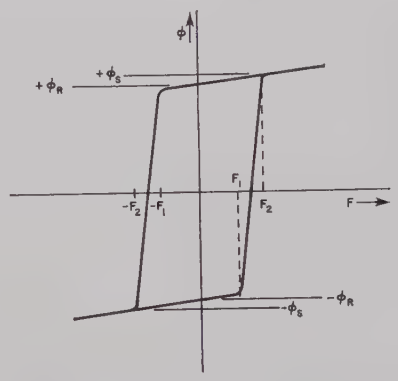


Fig. 1—Major loop properties.

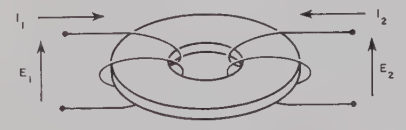


Fig. 2—Toroid with windings.

Fig. 3—Toroid terminology.

Terminology:

A core is said to be in the ONE or SET state when the flux is saturated in the clockwise direction.

A core is said to be in the ZERO or RESET or CLEARED state when the flux is saturated in the counterclockwise direction.

A "dot" is placed near the end of the winding which comes up towards the reader through the center of the core.
Windings are shown as a single turn.
When magnetic structures other than toroids are discussed, the dot notation and single turns are still used, but the information state is best described for the individual case.

The use of this dot terminology results in the following analysis rules:

Drive Conditions:

Current into a dot-end of a winding tends to set the core to the ONE state. During the period that the core is switching, this current sees a relatively high impedance. When the core reaches the ONE state, the impedance is relatively low. Current into a non-dot end of a winding tends to set the core to the ZERO state. As in the case of positive current, the impedance is high while switching, but low when the ZERO state is reached.

Induced Conditions:

When a core is switching towards the ONE state, the induced voltage on every winding on that core tends to place the dot end of the winding at the higher potential. Induced current has a tendency to flow out of all dot ends, opposing the switching of the core. When a core is switching towards the ZERO state, the dot ends go negative and induced current tends to flow into them.

A note on Mirror Notation: The increasing use of structures with more complex topology makes it difficult to use the standard mirror notation which has become popular for magnetic core toroids. A core and its equivalent mirror symbol are shown in Fig. 4. Current into a terminal is "reflected" by the mirror to show the tendency of the resulting flux vector to set the core to ONE or to ZERO. The directions of induced currents are then determined by "refracting" the changing flux vector through the mirror symbols. Since this terminology distorts the topology of the structure, it is not easy to use with multiaperture devices. In the interest of a uniform notation, therefore, the dot symbol rather than mirror symbolism is used in this survey.

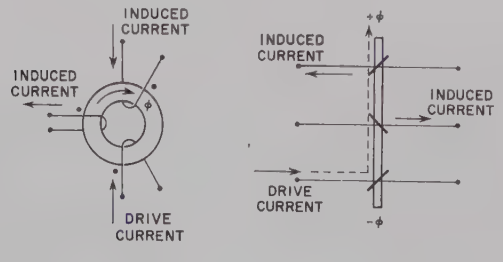


Fig. 4—A toroidal core and its mirror equivalent.

III. Square-Loop Magnetic Logic Circuits

The task of performing general digital logic requires circuitry capable of being arranged to provide any Boolean output function of a set of input variables. A complex function is usually developed step by step by passing the information through a succession of simple logic gates such as OR gates, AND gates, negation, etc. The ability to interconnect these gates with a reasonable amount of freedom implies the following circuit attributes of a logic scheme:

A bistable gain curve—The ONE and ZERO signals must remain identifiable in spite of circuit variations and random noise signals.
Unilateral information transfer—Information must flow from the variable inputs towards the output function.
Fan-in—The ability to form logical operations implies the nontrivial mixing of input signals.
Fan-out—In building up a complex logic function from simple functions, it is required that some intermediate variables appear at more than one point. A minimum fan-out of two is required to permit branching of signals.

To meet all these requirements with magnetic circuitry is not easy. Since the magnetic materials used are bilateral, providing unilateral circuit properties requires the use of semiconductors or a clever combination of geometry and the nonlinear properties of the magnetic materials.

The discussion to follow will be divided under four major headings:[1]

A. Core-Diode Schemes
B. Core-Transistor Schemes
C. Core-Capacitor Scheme
D. All-Magnetic Schemes

A. Core-Diode Schemes

The group of schemes using toroid cores and semiconductor diodes can be classified by the type of transfer circuit used, series or parallel.

Series-Circuit Transfer: This family of circuits has grown from the Ramey half-wave magnetic amplifier circuit [9]. As the name implies, this transfer circuit places the transmitting core S_n and receiving core S_{n+1} windings in series with the power source (Fig. 5). The receiver is assumed to be in the ZERO state. A fixed amount of volt-seconds is applied at the terminals, tending to drive the transmitter to the ZERO state and the receiver to the ONE state. If the transmitter is already

[1] It is to be noted that this survey is restricted to logic schemes that use the square-loop properties of magnetic materials as a major feature of the circuit. This survey does not include schemes that use only the nonlinear properties of magnetic cores, such as the parametron and various ferroresonance schemes. Moreover, no attempt has been made to include schemes in which a magnetic device plays a secondary role, *e.g.*, a pulse transformer, used in what is fundamentally a transistor logic scheme.

in the ZERO state, it switches further into saturation and presents a small impedance in the loop, and almost the full voltage is applied to the receiver. The loop current rises to a large value and the receiver is switched to the ONE state.

If the transmitter is in the ONE state, both cores will tend to switch. The turns ratios are adjusted so that the loop current required to switch the transmitter is below the receiver threshold current. The applied volt-seconds are almost entirely "absorbed" in the resetting of the transmitter, and the receiver cannot be set.

At the termination of transfer, the receiver is in the ONE state if the transmitter was in the ZERO state before transfer. If the transmitter had been in the ONE state, the receiver is now in the ZERO state, *i.e.*, if the variable x was stored in S_n, $\bar{x}$ is now in S_{n+1}.

The transmitter S_n is now in the ZERO state and can be used to receive the information from core S_{n-1} (Fig. 6).

Similarly, S_{n+1} can now be used as a transmitter to shift to S_{n+2}. The mechanism of transfer in the two loops $(n-1 \rightarrow n)$ and $(n+1 \rightarrow n+2)$ will be the same as just described for $(n \rightarrow n+1)$. Let us now consider the effect these transfers will have on the winding in loop $(n \rightarrow n+1)$.

As shown in Fig. 6, the effects of the induced voltages from both S_n and S_{n+1} are in a direction to cause a clockwise current to tend to flow, opposing the switching of cores S_n and S_{n+1}. Therefore, loop $(n \rightarrow n+1)$ must be open-circuited during the time the loops adjacent are transferring. A convenient way to accomplish this is by using sine wave sources for E. Phased as in Fig. 7, they will tend to oppose the induced voltages in the loops which are not transferring. A diode placed in the loop will insure that no reverse (counter-clockwise) currents flow. The basic Ramey amplifier circuit which results is shown in Fig. 7. To provide logic capability one may add the outputs of two transmitters, as in Fig. 8. Note that in Fig. 8(a) no path is provided for S_1 to reset in the event that S_2 does not switch, and vice versa. A path is usually provided with resistor decoupling or with current clippers. Furthermore, since this is basically a two-clock system, a non-negating transfer unit is needed to allow general logic. This may be a transistor flip-flop, a parallel transfer unit (see next section), or a three-clock positive series-transfer unit. (See Fig. 11.) Some fan-out circuitry for series transfer loops is shown in Fig. 9(a) and (b).

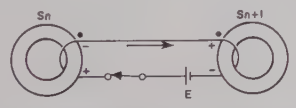


Fig. 5—Series-circuit transfer loop.

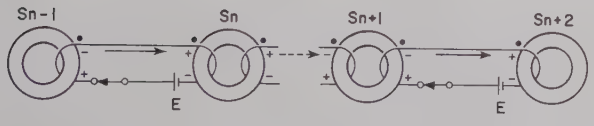


Fig. 6—A chain of series circuit loops.

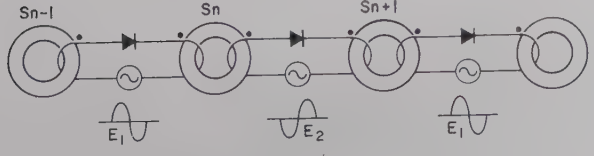


Fig. 7—Basic Ramey series transfer circuit.

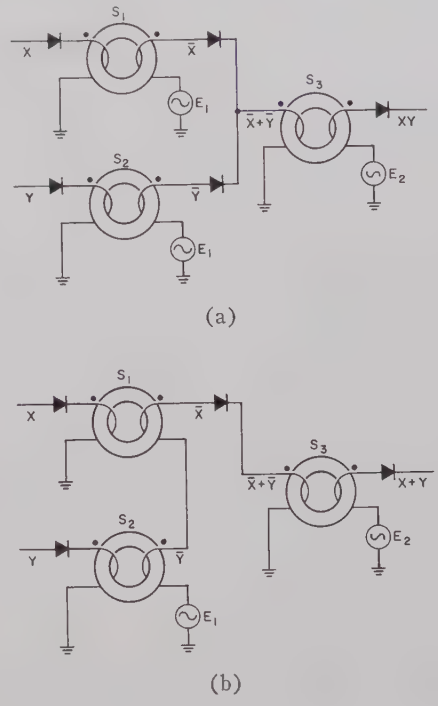


Fig. 8—Series circuit fan-in connections. (a) Outputs in parallel. (b) Outputs in series.

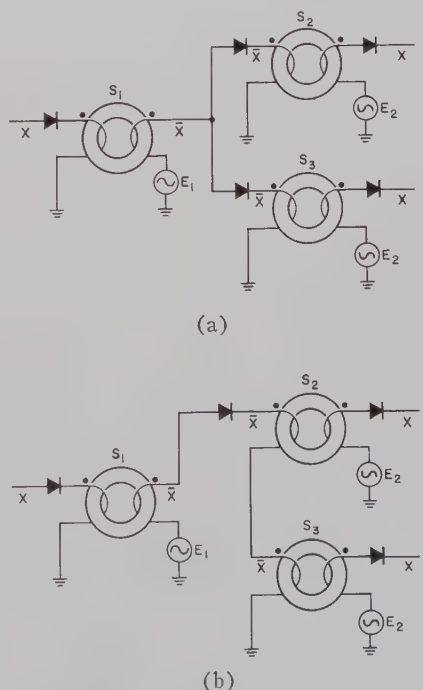


Fig. 9—Series circuit fan-out connections. (a) Inputs in parallel. (b) Inputs in series.

Notice that in Fig. 9(a) the receiver core which saturates first will rob current from the other. To prevent this, resistors and current clippers are usually added to the basic circuit.

For some commercial realizations of the Ramey series-circuit transfer, see Figs. 10–13. For simplicity, only a simple transfer circuit is shown in each case.

Parallel-Circuit Transfer: The parallel-circuit family is descended from the static magnetic shift register published by An Wang and Way Dong Woo [16] in 1950 (see Fig. 14). In this circuit, the transmitting core can be used as a transformer as well as an impedance. Like the Ramey circuit, the Wang-Woo circuit is a two-phase system in which, statically, alternate cores either carry information (available as transmitters) or are in the ZERO state (available as receivers). Transfer current tends to reset S_n to ZERO. If S_n switches, a current is induced in the loop, flowing through CR_n in a direction to set S_{n+1} to the ONE state. If S_n cannot switch, S_{n+1} is not switched. Thus S_{n+1} assumes the state that S_n had before the application of transfer current I. CR_{n+1} blocks the flow of induced current out of the S_{n+1} output when S_{n+1} is being set. CR_{n-1}, however, does not block the backward-going (clockwise) current induced in the input winding of S_n. The turns ratio of transmitters to receivers must be adjusted to cause this back voltage pulse to be small. The low-level pulse has its flux absorbed in the forward resistance of CR_{n-1} at a current below the switching threshold of S_{n-1}. The nonlinear forward characteristic of the diodes plays a major role in discriminating against low-voltage pulses.

Alternatively, this back current can be shunted from flowing back to S_{n-1} by using a resistor and a second diode in shunt (CR_s) (Fig. 15). R is picked large enough that the back-induced current does not retard too much the resetting of S_n, yet small enough to allow the forward-going current to switch S_{n+1}.

Another circuit can be derived from the Wang-Woo circuit which prevents back currents from flowing [17] (Fig. 16).

First consider the transfer of a data bit. In the new circuit, transfer current I (dotted line) is applied directly to the output winding. A loop current I_L flows, its magnitude such that the same amount of flux switches in both transmitter and receiver. If S_n is at ZERO, only a small elastic flux can switch. I_L is low, and S_{n+1} does not switch appreciably. If S_n contains a ONE, transfer current switches it to ZERO. I_L will be large and S_{n+1} is switched to the ONE state. The operation as described is quite similar to the Wang-

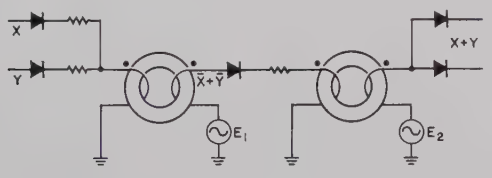
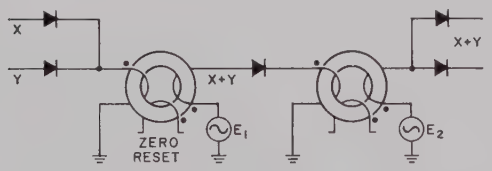
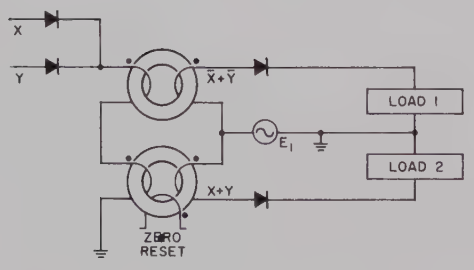


Fig. 10—The Logimag circuit. In the Logimag scheme [10], resistors are added to limit the current when switching of the receiver is complete and to insure that the transmitters can all reset.



(a)



(b)

Fig. 11—The Librascope Decision element. In the Decision element [11], it is not necessary to sample the state of the transmitter by driving it to the ZERO state. (a) Positive transfer. Transfer voltage E_1 links the transmitter and receiver so as to drive them both to the ONE state. The receiver is set to ONE only if the transmitter is already in the ONE state. A reset clock which sets the transmitter to ZERO is now needed. (b) Universal decision element. Used in combination with a second core driven as a standard Ramey circuit, the positive transfer technique results in a transmitting element which provides both a variable and its complement at every gate.

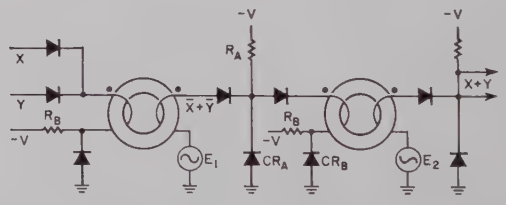


Fig. 12—The Westinghouse Cypak Inverting Circuit [12]. Two current sources have been added to the basic Ramey circuit: R_a, CR_a clips the output current, minimizing "ZERO" noise, and R_b, CR_b limits current on completion of switching.

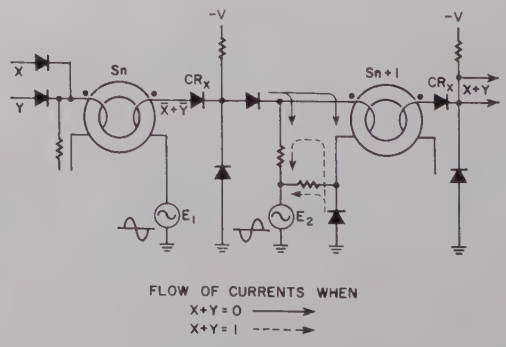


Fig. 13—The Remington-Rand Ferractor inverting circuit [13]–[15]. This circuit uses sine-wave current-limiting and adds a ZERO bias drive E_2 which clamps S_{n+1} to ZERO when E_1 is absorbed in the transmitter ($x=1$). This ZERO-drive allows the transmitter to be incompletely reset on readout, without false operation on the next readout. Furthermore, this allows operation at speeds where diode hole-storage is not negligible, and diode CR_x would pass unwanted reverse currents.

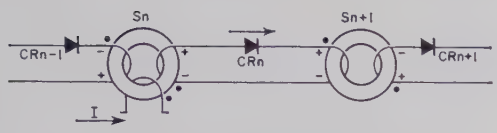


Fig. 14—Parallel circuit transfer.

Fig. 15—Alternate parallel transfer circuit.

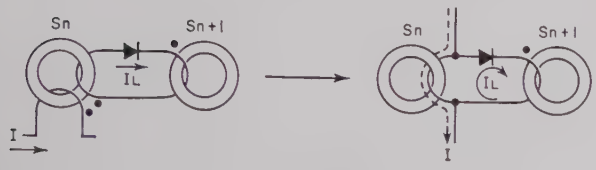


Fig. 16—Derivation of equivalent transfer loop.

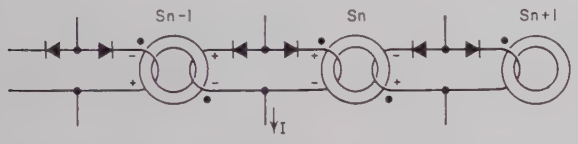
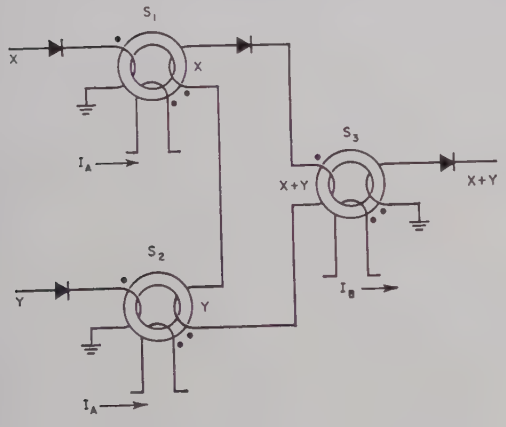
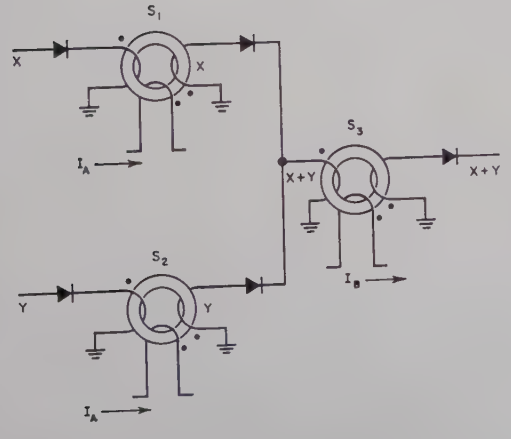


Fig. 17—Equivalent two-diode loop.



(a)



(b)

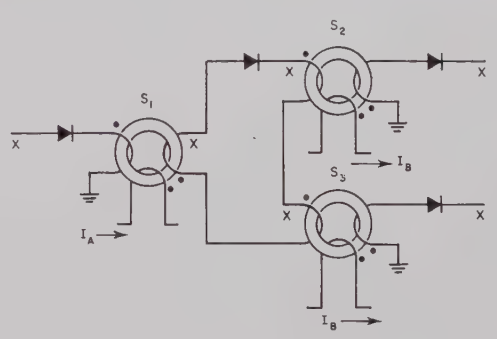
Fig. 18—Parallel-circuit fan-in connections. (a) Outputs in series. (b) Outputs in parallel.

Woo circuit. In the equivalent circuit, however, a second diode may be introduced to prevent false reverse currents from flowing. Since the *net* current in each winding is into the top line, diodes can be placed back-to-back at the top loop node as in Fig. 17.

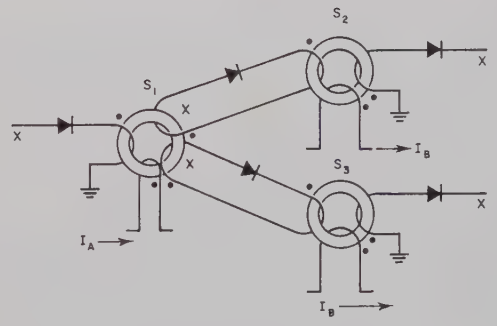
If transfer current I is not present in a loop, that loop is open-circuited to both directions of current flow. Reverse currents back from the transmitter and "leap-frogging" currents out of the receiver are prevented.

Some fan-in, fan-out circuits for parallel transfer are shown in Figs. 18 and 19.

Some means of providing logical negation is necessary to allow the parallel logic circuit to perform general logic. One solution is to use it in combination with a series circuit as in the Ferractor scheme [13]. It is also possible to develop a negative transfer as in Fig. 20



(a)



(b)

Fig. 19—Parallel-circuit fan-out connections. (a) Inputs in series. (b) Inputs in parallel.

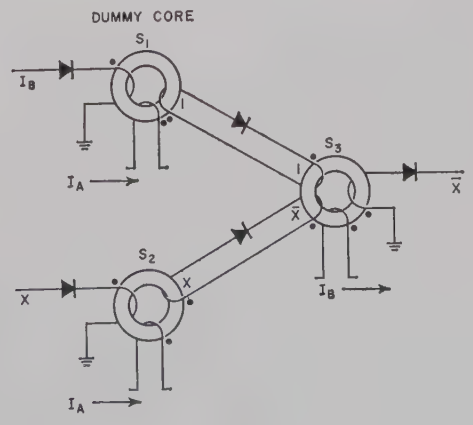


Fig. 20—Negation with parallel transfer circuit.

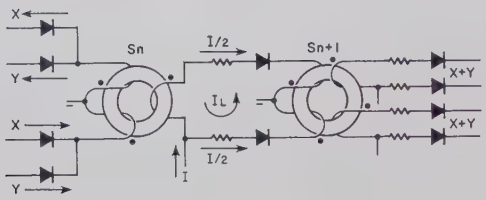


Fig. 21—Burroughs split winding circuit [17]. When the transmitter S_n does not switch, the equal branch resistances split transfer pulse I equally, and the net ampere-turns through the receiver are ZERO ($I_L=0$). The receiver responds only to loop currents generated by transmitter switching. Fan-in is at the diode cathodes. Fan-out is by separate output windings.

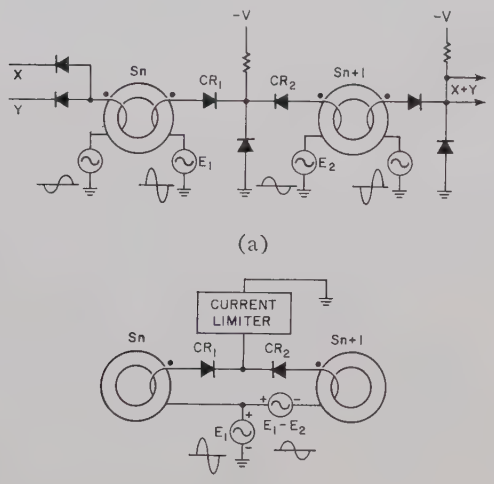


Fig. 22—Westinghouse Cypak noninverting circuit. (a) Actual circuit. (b) Equivalent model. In the Cypak circuit [12], drive is from a current-limited sine-wave voltage source rather than a current source. Transfer is initiated when E_1 and E_2 are positive going. The equivalent circuit shows that, effectively, a bias voltage is added which back biases CR_2 and prevents current from flowing through the receiver unless the induced EMF in the transmitter output winding is greater than E_1-E_2. If S_n contains a ZERO, it will not switch and CR_2 will be back biased. S_{n+1} will not be switched, remaining at ZERO. If the transmitter S_n contains a ONE, both cores will begin switching, with total current flowing through the limiter below its limiting point. The voltage drop across the output winding will be E_1, and the net voltage driving the receiver will be $E_1+(E_2-E_1)=E_2$. Fan-in is through diode mixing (additional diodes CR_2) at the input winding. Fan-out is from the cathode end of CR_1.

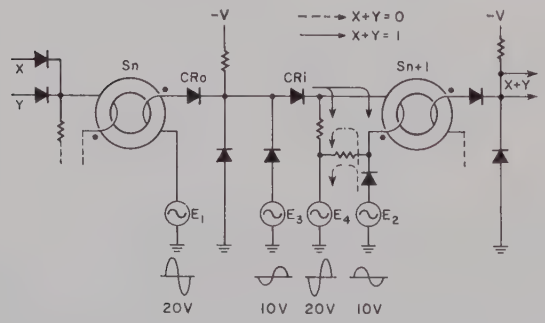


Fig. 23—Remington-Rand's Ferractor noninverting circuit [14], [15]. E_1 and E_2 serve the same roles as in the Cypak. E_3 adds 10 v back bias on CR_0 when the loop is not transferring. E_4 back biases CR_i when the loop is not transferring. In addition, E_4 clamps CR_i to ground when $x=1$ and allows E_2 to switch S_{n+1} to ONE. Note that the receiver ZERO state is re-established by E_1-E_2 if the transmitter is not set.

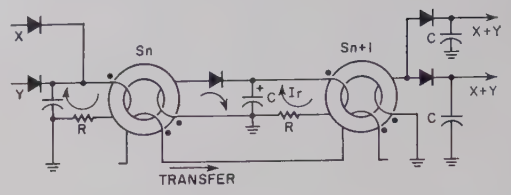


Fig. 24—Raytheon delay circuit [19]–[22]. A resistor-capacitor delay-network is added to the loop. Initially, every core contains information. Application of the transfer pulse switches all cores that are initially ONE to ZERO with a resultant charging of the associated capacitors. Subsequent discharge of these capacitors switches the neighboring cores through resistor R.

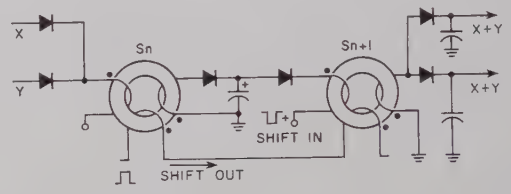


Fig. 25—RCA delay circuit [23]. This is similar to the Raytheon circuit (Fig. 24), but here the capacitor discharge loop is opened during the application of the shiftout pulse by back biasing diode CR_2. Application of the shift-in pulse will then allow the flow of discharge currents in those capacitors storing ONEs.

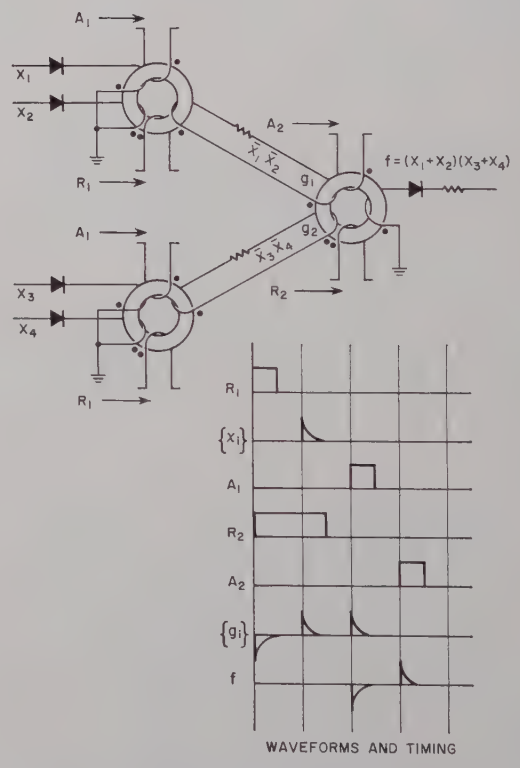


Fig. 26—Minnick's core-diode scheme. Circuits which reduce the number of diodes required for parallel transfer were developed by Minnick [24]. A typical circuit and its timing waveforms are shown in this figure. Diodes are needed in only one of the two levels of transfer circuits. With clever connections of circuits with fan-ins of (say) four, about seven out of eight of the diodes used in conventional circuits can be eliminated. However, the relative timing required is such that a recirculating pulse (a logic loop) must pass through three such two-level circuits, making six clocks required in all.

[18]. Referring to Fig. 20, I_B sets a ONE in the dummy core S_1, and variable X is stored in S_2. The ONE stored in S_1 is transferred to S_3 unless X is a ONE. If X is a ONE, it inhibits the pulse from the dummy core and a ZERO is stored in S_3.

Some commercial realizations of the parallel transfer family of schemes are shown in Figs. 21–26.

B. Core-Transistor Schemes

A group of schemes has been developed which uses magnetic structures in combination with active gain devices, such as transistors. In some cases, an active element is associated with each core; in others, one transistor flip-flop can be used to drive twenty or thirty magnetic elements.

1) *Transistorized Core-Diode:* In most of the core-diode schemes discussed in the previous section, the diodes can profitably be replaced by suitable connections of transistor emitter-followers or inverters. The transistor can increase margins, fan-out isolation, and drive sensitivity. For examples of this, see Hoffman and Maclean [25], NBS Tech. RPT-2384 [26], Perry Hoffman and Shallow [27].

2) *Regenerative Transistor-Core:* A transistor-core combination with the square-loop core providing feedback can be operated as a bistable blocking oscillator (see Fig. 27) [28]–[31]. Application of a triggering pulse induces current in the transistor base and collector circuits which regeneratively switch the core to the ZERO state. The current in the collector, if a ONE was stored in the core, can be used to set other cores. The use of a delay unit between stages allows one core per bit and a single clock.

A variation of this scheme has been used in a shift register of the RCA Megacoder paging system (Fig. 28). Here a capacitor supplies the energy to read-out cores and to transfer (as the capacitor recharges) the information to the next core [32], [33].

3) *Core-Logic Transistor Flip-Flop:* Many magnetic schemes use magnetic cores for logic and transistor (or tube) circuits such as flip-flops for power gain and standardization. The early work in this field by M. K. Haynes [34] and E. A. Sands [35] was of this nature.

Inhibit Technique: It is possible to construct core logic such that no input logic line must switch flux. Instead, the logic lines are used to inhibit the switching of flux. This is a generalization of the trick used to obtain negation in the parallel transfer core-diode schemes (Fig. 20) [18]. To insure adequate overlap of inhibit pulses and set pulses, flip-flops or pulse stretchers are used in this technique to establish the inhibiting currents.

a) *National Cash Register's inhibit core logic* [36]: In Fig. 29, a negative clock pulse on the vertical set-drive winding sets Core S_1 unless it is inhibited from switching by current from flip-flops A or B. Similarly, if not inhibited by C or D, Core S_2 is set. When the clock pulse reverses, it tends to reset both cores to ZERO. If either core had been set, its resetting now sets the next flip-flop. Since input inhibit lines never switch flux, one flip-flop can control many inputs. Logic gates of this type, but having a different topological form (Rod Logic), have also been developed at NCR [37].

b) *The Bell Laboratories Laddic* [38]: The Laddic uses a more complex topology than the simple core to achieve a class of logic functions (Fig. 30). The ladder-like struc-

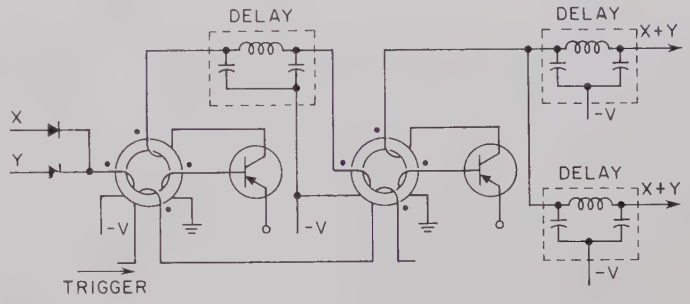


Fig. 27—Regenerative core-transistor scheme.

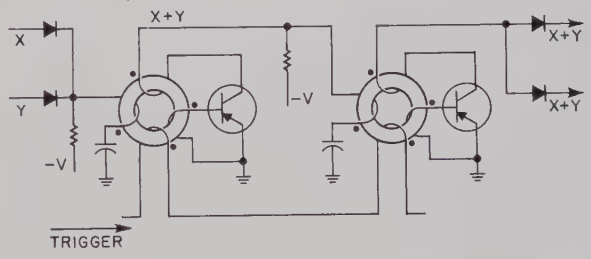


Fig. 28—RCA Megacoder scheme.

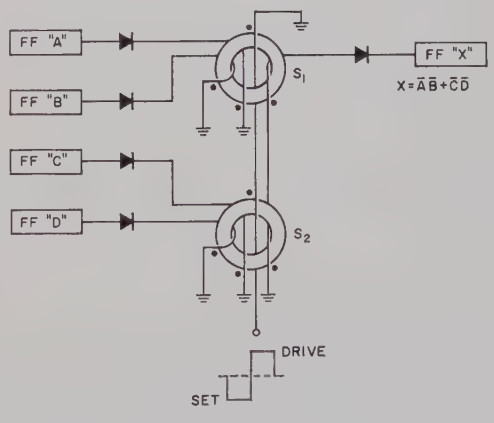


Fig. 29—National Cash Register's inhibit-core scheme.

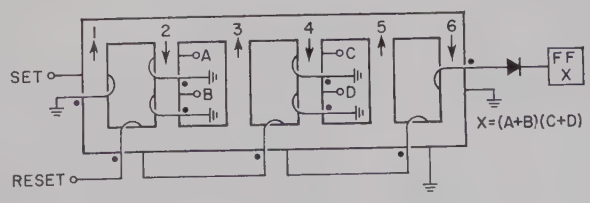


Fig. 30—Bell Laboratories' Laddic

ture has an even number of "rungs." Application of the reset pulse leaves the flux in the ladder rungs in the state pictured. When a clock-set pulse is applied, it switches flux downward in Leg 1. The lines of switched flux close by switching in Leg 2, unless inhibit currents from flip-flops A or B are present. If Leg 2 is inhibited, the flux attempts to close by switching Leg 4 (Leg 3 is already saturated—little flux can switch). If inhibit currents C or D are present, Leg 6 is the only leg through which the switched flux from Leg 1 can close. Upon re-application of the reset pulse, the output flip-flop receives a positive pulse if (A or B) and (C or D) inhibit currents are present. Notice that an output winding linking Leg 4 would receive an output if $(A+B)\times(\overline{C+D})$. Work at Bell Laboratories has allowed an extension of this Laddic-inhibit technique to synthesizing magnetic analogs or relay networks [39].

c) *The Aeronutronic Biax* [40]: The Biax magnetic-logic structure consists of a solid rectangular parallelepiped with open, intersecting tunnels along two axes (Fig. 31). To a good approximation, coupling between windings A and B can be shown to be the same as in an equivalent two-core circuit [41].

The operation of the Biax circuit (Fig. 32) can be understood from the equivalent core circuit. Windings threading the Biax couple to equivalent cores S_1 in the same manner as in the NCR inhibit technique (Fig. 29). The effect of S_2 is to cancel any ZERO noise that results from elastic switching of S_1.

4) *Eldridge's Inhibit Technique* [42]: An inhibit technique using a transistor clamp has been described by Eldridge (Fig. 33). Two pulse systems attempt to set and reset all cores, but a transistor connected across the input winding on each core prevents the core's being set (by presenting a short-circuit load) unless a turn-off signal is applied to the transistor base. The output winding signals can be added algebraically to provide *AND*, and *NEGATION*, as well as *OR* functions. (Since the saturated transistor is not a perfect short circuit, some flux is set into the core even when no input signal arrives. This "zero-flux" does not build up, however, since the reset completely clears the cores each clock cycle.)

5) *Miscellaneous Multi-Aperture Device, Flip-Flop Schemes:*

a) *IBM flux logic scheme* [43]: This scheme uses a structure grossly similar to the Laddic (Fig. 30), but has a different mode of operation. The reset state is as shown in Fig. 34. Notice that the "rails" of the ladder must be

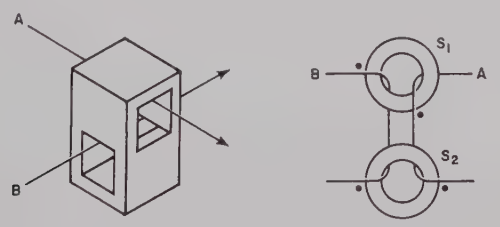


Fig. 31—The aeronutronic biax element and an equivalent circuit.

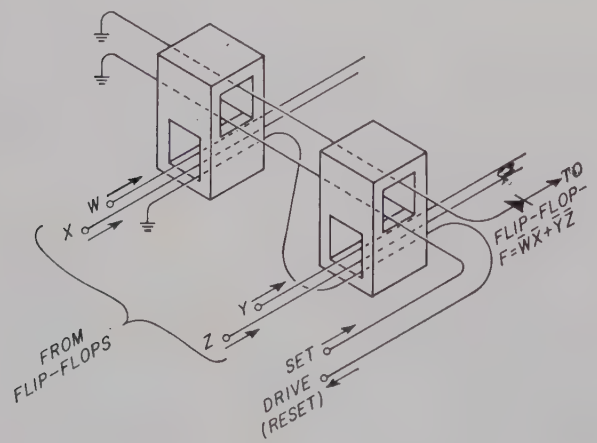
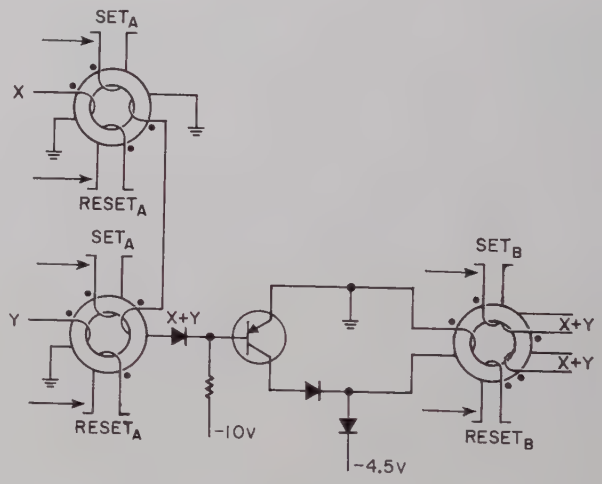


Fig. 32—Biax logic circuit.



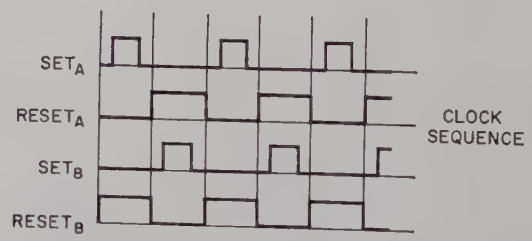


Fig. 33—Eldridge's inhibit transistor-core scheme.

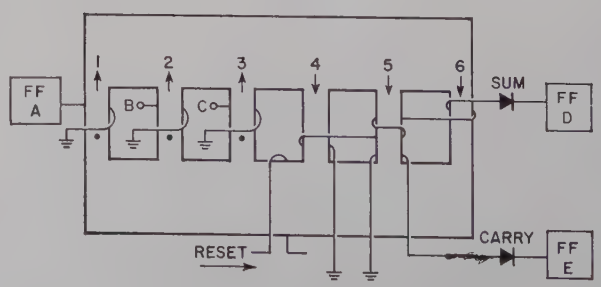


Fig. 34—IBM flux logic scheme.

wide enough to accommodate the flux from all the "rungs." The inputs on Rungs 1, 2, and 3 can reverse the flux in these rungs. Because flux tends to switch about the shortest available path, the flux in Rung 4 switches if one or more inputs are present; the flux in Rung 5 reverses if two or more inputs are present; three inputs present cause Rung 6 to reverse. Upon application of the reset pulse, windings threading the outputs sense the resetting of any rungs that switched. Fig. 34 shows two outputs coupled to form a sum and carry function.

b) *RCA transfluxor scheme* [44]: The transfluxor consists of a toroid with a small aperture in one side (Fig. 35). When the core is in the clear state, I_{ac} can switch flux in Leg 3 only by switching flux about the main aperture: when I_{ac} is positive, it tends to push L_3 further into saturation and little flux can switch; when I_{ac} is negative, L_3 can switch but L_2 is saturated, so L_3 can switch only by switching around the main aperture through L_1. By appropriately adjusting the amplitude of control current I, it is possible to put the transfluxor in a set state. I_{ac} can now switch flux around the output aperture, and an ac voltage is induced in the output winding. The magnitude of the output voltage will depend on how much of the flux in L_2 was set by I. The transfluxor will store this input setting through an indefinite number of cycles of I_{ac}. Using the reset and maximum set states as binary ONES and ZEROS, the transfluxor can be used to store digital information.

If multiple inputs are used to control the transfluxor and the output voltage level is detected and used to set a flip-flop, the transfluxor flip-flop combination is capable of general logic.

Multiple-hole transfluxors, or "logicors" have been developed to give additional logic flexibility to the transfluxor technique [45], [46].

C. *Société d'Electronique et d'Automatisme—Core-Capacitor Scheme* [47]

This scheme uses no semiconductors in the coupling loops. It requires three cores per bit and three clocks. The procedure is not to block the flow of unwanted currents (forward out of a receiving core or backwards out of a transmitter); instead of this, a system of clock clamps and capacitors is used to allow operation in spite of these spurious currents. Typically, information is stored in S_n; S_{n+1} and S_{n+2} are in the ZERO state. (Since each group of three cores is treated identically, S_{n-1} is also in the ZERO state. See Fig. 36.)

While current I_3 is flowing, transfer current I_1 is applied (tending to reset S_n to zero); if S_n contains a ZERO, no switching occurs. If S_n contains a ONE, capacitors C_{n-1} and C_n charge through saturated cores S_{n-1} and S_{n+1}, tending to drive both cores toward ZERO [Fig. 37(a)]. When S_n finishes switching, the capacitors send discharge currents in the reverse direction [Fig. 37(b)]. Since I_1 and I_3 are clamping S_n and S_{n-1} to ZERO, they cannot switch. C_{n-1} discharges quite quickly and causes no switching. Capacitor C_n switches S_{n+1} through C_{n+1} in a direction to set S_{n+2} to ONE. Since I_3 is still clamping S_{n+2} to the zero state, S_{n+2} presents a low impedance and capacitor C_n discharges quickly into C_{n+1} until they are at equal potentials. Then C_n and C_{n+1} both discharge into S_{n+1} in a direction to set it to ONE [Fig. 37(c)]. The information previously stored in S_n is now in S_{n+1}; and S_n, S_{n-1}, and S_{n+2} are in the ZERO state.

This discussion has demonstrated a shift-register technique. The extension of this method to logic is quite

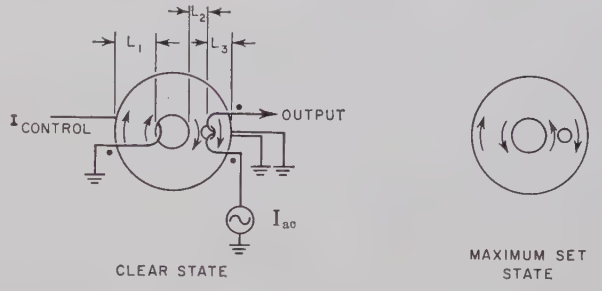


Fig. 35—RCA's transfluxor.

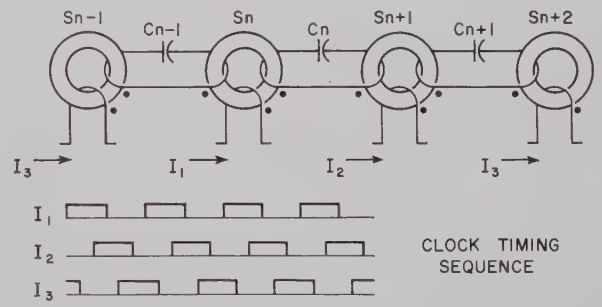


Fig. 36—Core-capacitor scheme.

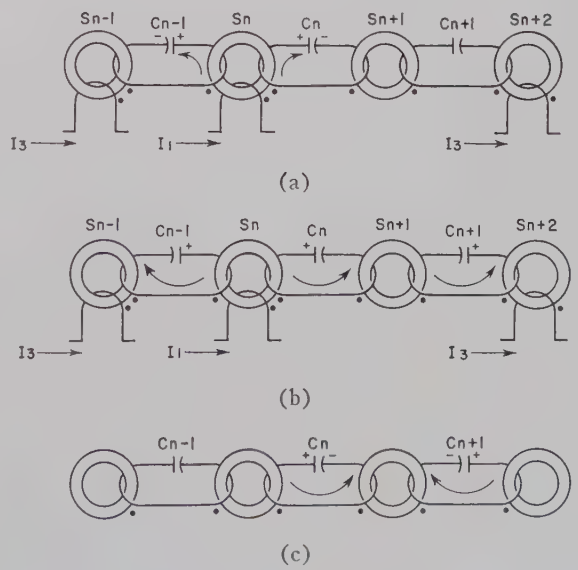


Fig. 37—Core-capacitor scheme transfer currents. (a) Resetting S_n, changing C_{n-1} and C_n. (b) C_{n-1} and C_n discharge, C_{n-1} charges. (c) C_n and C_{n+1} discharge setting S_{n+1} to one.

straightforward. Fan-in and fan-out are accomplished by adding additional coupling loops. The generation of OR, AND, and NEGATION are the same as in the core-diode parallel-transfer techniques shown in Fig. 18(b), 19(b) and 20.

D. All-Magnetic Schemes

There has been much interest in evolving magnetic logic schemes requiring no diodes or transistors, *i.e.*, coupling-isolation and gain completely handled by the magnetic elements and interconnecting wire. (See the bibliographical sketch on all-magnetic logic schemes on page 203 of this issue.) While some of the schemes included in this section require resistance in the coupling loops, the resistance is usually of such an order of magnitude that the wire in the loops can provide it.

1) *IBM Biased-Core Scheme, Type I* [48]–[50]: This scheme uses cores for the unilateral coupling elements in transfer loops (Fig. 38). Loop currents flowing counterclockwise tend to switch Cores C_T to the ONE state, with a resulting high-loop impedance. Loop currents flowing clockwise do not switch C_T, and a low-loop impedance is provided. This unilateral flow property of the cores allows them to replace the diodes usually used in the coupling loops. The reset current I_r is necessary to re-establish the ZERO state in all the C_T cores. The resetting of those C_T cores that had been set must be done slowly so that the induced voltage in the loop is small. The resulting loop current when resetting is kept below the threshold of the *S*-cores, and the flux stored in C_T is absorbed in the loop resistance, *R*. (This scheme is a direct analog of the Wang-Woo core-diode scheme, which depended on the diode threshold for prevention of back currents (Fig. 14). No corresponding threshold in the core equivalent of the diode is apparent. Also see comments in the bibliographical sketch elsewhere in this issue.)

2) *IBM Biased-Core Scheme, Type II* [48], [49]: This version of the biased-core scheme uses the coupling cores as transformers rather than as two-terminal impedances (Fig. 39). Coupling of signals through these transformers is controlled by clock currents, to allow controlled-signal propagation as follows: Assume all cores except S_n to be in the ZERO state. If S_n is also ZERO, application of clock currents has no effect. If S_n is in the ONE state, however, I_A switches it to ZERO. The resultant induced EMF on the other S_n winding will set C_{T2} and S_{n+1} to ONE (C_{T1} and C_{T3} are clamped to ZERO by I_A). I_R then slowly resets C_{T2} to ZERO, and the information in S_n has been transferred to S_{n+1}.

3) *The SRI Multi-Aperture Device (MAD)* [51–53]: The basic MAD element (Fig. 40) is a three-aperture ferrite structure somewhat similar to a transfluxor (Fig. 35). Using the three-hole geometry allows one MAD to control the state of others, using only clock pulses and interconnecting wire. Input-output isolation is possible since flux can be switched locally about one small aperture without affecting the other. Since the impedance level of the coupling loop does not have to be matched to a semiconductor characteristic, single turn coupling is possible.

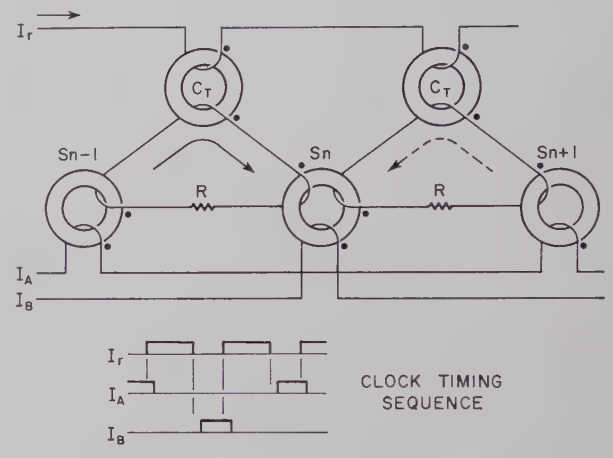


Fig. 38—IBM biased-core scheme, type I.

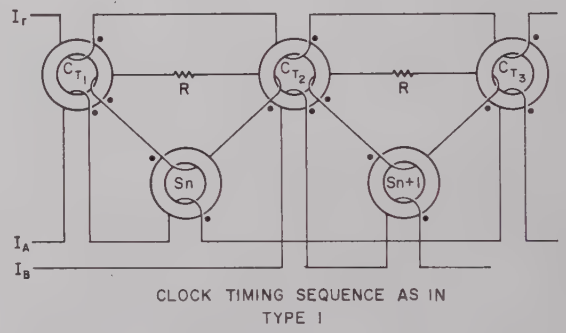


Fig. 39—IBM biased-core scheme, type II.

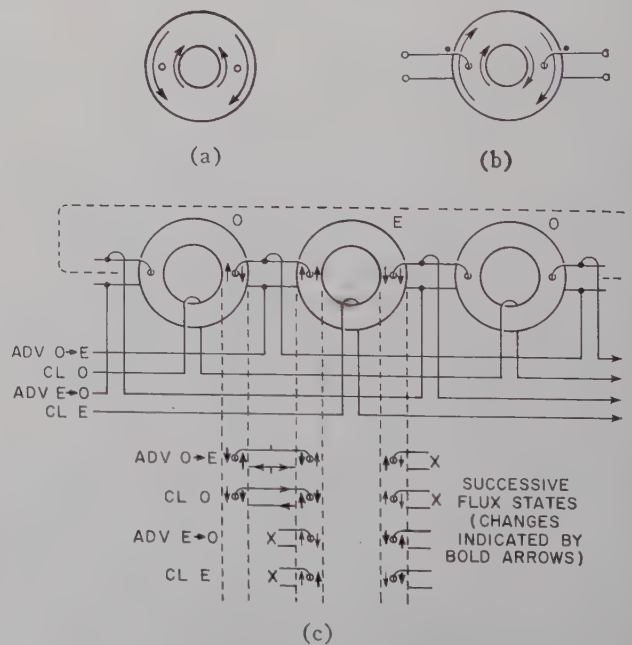


Fig. 40—The SRI MAD element. (a) Clear state. (b) Set state. (c) Shift-register clock cycle.

A MAD shift register is shown in Fig. 40(c). The operation is grossly similar to a parallel-transfer core-diode shift register (Figs. 14 and 15). Since the MAD readout is nondestructive, two additional clock cycles are needed to clear MADs and ready them for use as receivers. To follow a transfer cycle, assume the information is stored in the 0 elements. If the bits stored are ZEROs, clock currents have no significant effect. If a ONE is stored in the first 0 core, flux will switch at the application of each clock cycle as indicated by the heavy arrows under the transfer loops. Notice that transfer operations in one loop do not couple flux changes to other loops, so no isolating diodes are required. The extension of this shift operation to logic requires the same methods used in extending the generality of the core-diode shift registers. (See Figs. 18–20.) The same type of all-magnetic logic can be realized using arrays of simple toroidal cores in the place of each MAD element (Fig. 41). These toroidal arrays have been labeled "synthetic-aperture devices," or SADs [54]. The SADs are the realization of the equivalent magnetic circuits of the MADs in simple cores.

4) *The Multi-Aperture Device, Resistance (MAD-R) Scheme* [55], [56]: This scheme (Fig. 42) may be implemented with two-aperture MADs similar to transfluxors (Fig. 35), with I_p replacing I_{sc}. The initial flux state shown in Fig. 42 is that existing immediately after an I_B pulse has cleared S_{n-1} and S_{n+1} and transferred a ONE from S_{n-1} to S_n. The succeeding I_p pulse, which must be low enough not to set flux spuriously around any MAD in a cleared state, changes the state of S_n to that shown opposite the I_p pulse. The flux linked into the output coupling loop is absorbed in the loop resistance R. The loop current I_L is negative and hence cannot switch S_{n+1}. The new flux states of S_n and S_{n+1} are shown opposite the I_A pulse. The volt-seconds linked into the input loop of S_n at the same time are absorbed in resistance R, since I_A is passed through a winding linking the output aperture of S_{n-1} in order to prevent the accompanying positive current I_L from switching S_{n-1}. In this way, unilateral transfer is obtained for arbitrarily high values of I_A because of prevention of backward transfer by I_A itself. Note that S_{n+1} now stores the flux state originally held by S_n. Completion of one cycle of the clock sequence with another pulse of I_p followed by a pulse of I_B (not shown) transfers the ONE from S_{n+1} to S_{n+2}. The operation as described above is affected only in minor detail if the current I_p is steady (dc) rather than pulsating.

The currents I_A and I_B have higher tolerances than the transfer currents for the straight MAD scheme since they are not threshold-limited.

The current I_p is threshold-limited but may also have wider tolerance for low-speed operation since the lower limit is then determined only by the minor aperture threshold. As in the IBM schemes (Figs. 38, 39), the potential speed of operation is lower than for the straight MAD scheme. The toroidal-core equivalent (using SAD techniques) of the MAD-R scheme requires six cores per bit and hence does not appear to be the same as either of the IBM schemes. (The additional magnetic paths in MAD-R make it possible to obtain wide current-tolerances.)

5) *Thin-Film Logic*

a) *Coherent rotation scheme* [57]–[59]: Thin ferromagnetic films may show single-domain behavior. They possess an easy or preferential axis of magnetization, and thus exhibit a bistable magnetization characteristic. One small dot-film is capable of controlling at least one other by means of a gain mechanism based on the inherent, dynamic bistability of the film. In Fig. 43 the vertical dotted line represents the easy axis for the film spot. Current I_t induces a transverse field which can rotate the field vector M towards the horizontal. A small signal current I_s is then capable of controlling the direction of fall-back of the vector M. As M falls back, it induces an EMF in any conducting loop which is coupled to it, and this EMF can be used as a source for

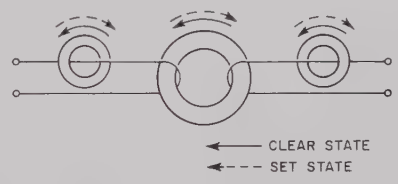


Fig. 41—The SRI SAD circuit.

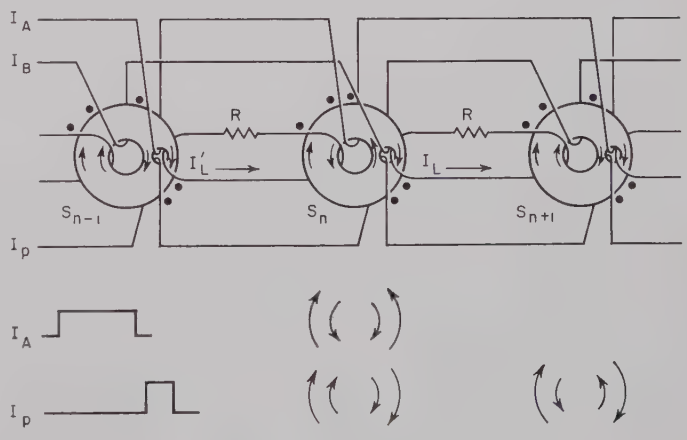


Fig. 42—The SRI MAD-R.

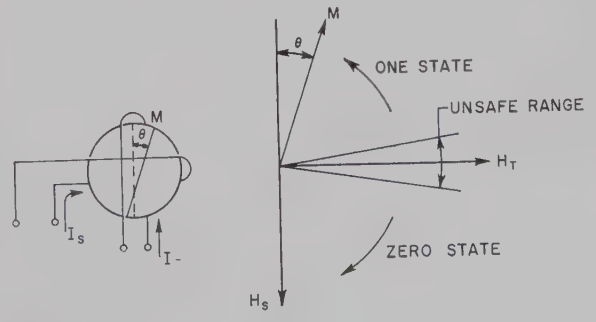


Fig. 43—Bistable thin film element.

I_s to control a second film spot. This gain mechanism may be used in various topologies to perform logic. It appears possible, for instance, to construct a scheme similar to the synthetic-aperture device (SAD) mentioned previously. Other simpler structures can be used which take advantage of the special dynamic form of bistability of the film dots to reduce the circuit to one dot element per stage.

b) *Domain stepping* [60], [61]: This technique is based on the fact that existing domains in thin films can be moved by fields that are not strong enough to create new domains. Shift registers using this principle have been successfully demonstrated, and a complete logic scheme has been hypothesized.

Conclusions

For reasons of speed, cost, reliability, ruggedness and increased component density, many new circuits and devices have been suggested to replace vacuum tubes and/or transistors in the logic and control sections of digital computers. One strong contender in this challenge is the class of square-loop magnetic circuits. It is hoped that this paper has helped introduce the reader to this promising circuit area.

Bibliography

[1] W. N. Papian, "Magnetic-Core Applications in Digital Computers," lecture in Special Summer Course in Digital Computers and Data Processors, pt. S.1-13 p., Univ. of Mich. Engrg. College, Ann Arbor; July, 1955.

[2] A. J. Lincoln, "Ferromagnetic Core Logical Circuitry and its Application to Digital Computers," Aberdeen, Md., BRL Rept. No. 911; August, 1955.

[3] J. A. Rajchman, "Magnetics for computers—a survey of the state of the art," *RCA Rev.* vol. 20, pp. 92–135; March, 1959. See also *Proc. Special Tech. Conference Non-linear Magnetics and Magnetic Amplifiers*, Los Angeles, Calif., pp. 179–221; August, 1958.

[4] E. G. Clark, "A Survey of Solid-State Logic Circuitry," *Digest of Tech. Papers*, 1958 Transistor and Solid-State Circuits Conference, Philadelphia, Pa., p. 49; February 20–21, 1958.

5] E. C. Orris, "Literature Survey on Solid-State Logic Elements," WADC Sc. Rept. No. 848-1, Contract AF 33(616)-5923, Ohio State Univ. Res. Foundation, Columbus, Ohio; January, 1959; AD 211058.

[6] A. J. Meyerhoff, "Digital Magnetic Circuits, Theory, Electrical Design and Logical Design," Tech. Rept. No. TR 59-27, Burroughs Corp., Contract No. 2145(00); April, 1959; AD-216961, AD-215789, AD-215790.

[7] D. H. Looney, "Recent advances in magnetic devices for computers," *J. Appl. Phys.*, suppl. to vol. 30, pp. 38S–42S; April, 1959.

[8] M. Karnaugh, "Pulse switching circuits using magnetic cores," Proc. IRE, vol. 5, pp. 570–584; May, 1955.

[9] R. A. Ramey, "The single-core magnetic amplifier as a computer element," *Elec. Commun.*, vol. 4. pp. 442–446; January, 1953.

[10] J. Auricoste and H. Lhomme, "Les circuits logiques magnetiques," *Onde elect.*, vol. 38, pp. 217–224; March, 1958.

[11] R. K. Richards, "Digital Computer Components and Circuits," D. Van Nostrand Co., Inc., Princeton, N. J., 1957, pp. 240–249.

[12] R. K. Richards, *op. cit.*, pp. 249–252.

[13] T. H. Bonn, "The Ferractor," *Conf. on Magnetism and Magnetic Materials*, Boston, Mass., Vol. T-91, pp. 654–665; October, 1956.

[14] T. H. Bonn, "The megacycle ferractor," 1956 IRE WESCON Convention Record, pp. 118–119.

[15] R. D. Torrey and T. H. Bonn, "A 2.5 mc ferractor accumulator," *Proc. Eastern Joint Computer Conf.*, pp. 50–53; December, 1956.

[16] A. Wang and W. D. Woo, "Static magnetic storage and delay line," *J. Appl. Phys.*, vol. 21, pp. 49–54; January, 1950.

[17] D. Loev, W. Miehle, J. Paivinen and J. Wylen, "Magnetic core circuits for digital data processing systems." Proc. IRE, vol. 44, pp. 154–162; February, 1956.

[18] N. B. Saunders, "Magnetic binaries in the logical design of information handling machines," *J. ACM*, pp. 223–229; May, 1952.

[19] R. D. Kodis, "Application and performance of magnetic-core circuits in computing systems," *Proc. Eastern Joint Computer Conf.*, pp. 30–34; 1954.

[20] S. Guterman, R. D. Kodis and S. Ruhman, "Logical and control functions performed with magnetic cores," Proc. IRE, vol. 43, pp. 291–298; March, 1955.

[21] S. Guterman, R. D. Kodis, and S. Ruhman, "Circuits to perform logical and control functions with magnetic cores," 1954 IRE Convention Record, pt. 4, pp. 124–132.

[22] R. D. Kodis, S. Ruhman and W. D. Woo, "Magnetic shift register using one core per bit " 1953 IRE Convention Record, pt. 7, pp. 38–42.

[23] V. L. Newhouse and N. S. Prywes, "High speed shift registers using one core per bit," IRE Trans. on Electronic Computers, vol. EC-5, pp. 114–120; September, 1956.

[24] R. C. Minnick, "Magnetic Switching Circuits," Ph.D. thesis, Harvard University, Cambridge, Mass.; April, 1953, AD 72241, (also *J. Appl. Phys.*, vol. 25, pp. 479–475; April, 1954).

[25] G. R. Hoffman and M. A. Maclean, "Quiescent core-transistor counters," *Proc. IEE*, vol. 103, pt. B, suppl. 3, pp. 418–421; April, 1956.

[26] "Improved Magnetic Core Digital Circuitry," U. S. Dept. of Commerce, Natl. Bur. Standards, STR-2384; September, 1959.

[27] G. H. Perry, G. R. Hoffman, and E. W. Shallow, "A new and simple type of digital circuit technique using junction transistors and magnetic cores," *Proc. IEE*, vol. 103, pt. B, suppl. 3, pp. 412–417; April, 1956.

[28] A. Wang, "Static magnetic memory—its applications to computers and controlling systems," *J. ACM*, pp. 207–212; May 2–3, 1952.

[29] S. S. Guterman and W. M. Carey, Jr., "A transistor-magnetic core circuit; a new device applied to digital computing techniques," 1955 IRE Convention Record, pt. 4, pp. 84–994.

[30] J. Porter, "Six ways to use magnetic shift register elements," *Electronics*, pp. 80–81; January 15, 1960.

[31] D. Eldridge, "Discussion on 'The transistor as a computing element—II' " *Proc. IEE*, vol. 103, pt. B, suppl. 3, pp. 422–424; April, 1956.

[32] H. Kihn and W. E Barnette, "The Megacoder—a high-speed, large-capacity micro-miniature decoder for selective communication," *RCA Review*, vol. 20, pp. 153–179; March, 1959.

[33] F. Rozner and P. Pengelly, "Transistors and cores in counting circuits," *Electronic Engrg.*, vol. 31, pp. 272–274; May, 1959.

[34] M. K. Haynes, "Magnetic Cores as Elements of Digital Computers," (TR to ONR) Ph.D. dissertation, Univ. of Illinois, Urgana, Ill.; August, 1950.

[35] E. A. Sands, "Magnetic Core Switches," Progress Repts. on AF 33(600)24925 by Magnetics Res. Inc., AD 39072, AD 22479; June–December 1953. WADD-PRI-AD 38072, June–September, 1953; WADD-PR2-AD 22479, September–December, 1953.

[36] L. J. Andrews, "A technique for using memory cores as logical elements," *Proc. Eastern Joint Computer Conf.*, pp. 39–46; December, 1956

[37] D. A. Meier, B. A. Kaufman, and D. W. Rork, "Megacycle magnetic rod logic," 1959 IRE WESCON Convention Record, pt. 4, pp. 27–31.

[38] U. F. Gianola and T. H. Crowley, "The laddic—a magnetic device for performing logic," *Bell Sys. Tech. J.*, vol. 38, pp. 45–72; January, 1959.

[39] D. B. Armstrong, T. H. Crowley, V. F. Gianola, and E. E. Newhall, "Magnetic analogues of relay contract networks," IRE Trans. on Electronic Computers, vol. EC-9, pp. 30–34; March, 1960.

[40] C. L. Wanlass and S. D. Wanlass, "Biax high speed magnetic computer elements," 1959 IRE WESCON Convention Record, pt. 4, pp. 40–54.

[41] Private communication with D. Engelbart of Stanford Research Institute, Menlo Park, Calif.

[42] D. Eldridge, "A new high-speed digital technique for computer use," *Proc. IEE*, vol. 106, pt. B, pp. 229–236; March, 1959. (See also discussion pp. 237–239.)

[43] N. F. Lockhart, "Logic by ordered flux changes in multipath ferrite cores," 1958 IRE National Convention Record, pt. 4, pp. 268–278.

[44] J. A. Rajchman, and A. W. Lo, "The transfluxor," *RCA Rev.*, vol. 16, pp. 303–311, June, 1955. Proc. IRE, vol. 44, pp. 321–332; March, 1956.

[45] L. S. Onyshkevych, "Analysis of Circuits with Multi-Hole Magnetic Cores," Tech. Rept. 329, M.I.T., Cambridge, Mass., July, 1957. Thesis, Research Lab. of Electronics, M.I.T., DA-36-039 SC-64637.

[46] H. W. Abbott and J. J. Suran, "Multihole ferrite core configurations and applications," Proc. IRE, vol. 45, pp. 1081–1093; August, 1957.

[47] M. Dumaire, H. Jeudon, and M. Lilamand, "Logical circuits using magnetic cores and capacitors," *Proc. of Special Tech. Conf. Nonlinear Magnetics and Magnetic Amplifiers*, Los Angeles, Calif., pp. 243–266; August, 1958.
[48] L. A. Russell, "Diodeless magnetic core logical circuits," 1957 IRE Convention Record, pt. 4, pp. 106–114.
[49] E. Bloch and R. C. Paulsen, "Magnetic core logic in a high-speed card-to-tape converter," IRE Trans. on Electronic Computers, vol. EC-8, pp. 169–181; June, 1959.
[50] T. J. Rey, "Registers such as are employed as digital computing apparatus," Patent Application filed May 28, 1952; granted July 13, 1954; U. S. Patent No. 2,683,819.
[51] H. D. Crane, "A high speed logic system using magnetic elements and connecting wire only," *Proc. Special Tech. Conf. on Nonlinear Magnetics and Magnetic Amplifiers*, Los Angeles, Calif., pp. 465–482; August, 1958. Proc. IRE, vol. 47, pp. 63–73; January, 1959.
[52] N. S. Prywes, "Diodeless magnetic shift registers utilizing transfluxors," IRE Trans. on Electronic Computers, vol. EC-7, pp. 316–324; December, 1958.
[53] D. R. Bennion and H. D. Crane, "Design and analysis of MAD transfer circuitry," *Proc. Western Joint Computer Conf.*, pp. 21–36; March 3–5; 1959.
[54] D. C. Engelbart, "A new all-magnetic logic system using simple cores," *Digest of Tech. Papers*, 1959 Solid-State Circuits Conf., pp. 66–67.
[55] D. R. Bennion, "A note on magnetic shift registers," IRE Trans. on Electronic Computers, vol. EC-9, p. 262; June, 1960.
[56] U. F. Gianola, "Integrated magnetic circuits for synchronous sequential logic machines," *Bell Sys. Tech. J.*, pp. 295–332 March, 1960.
[57] D. C. Engelbart, "High Speed Components for Digital Computers," Stanford Res. Inst., Menlo Park, Calif., Quart. Repts. on AF 33(616)-5804; June, 1958 to November 1959. AD 303578, AD 211620, AD 216749.
[58] M. W. Green, "High Speed Components for Digital Computers," Stanford Res. Inst., Menlo Park, Calif. Final report on AF 33(616)-5804; December, 1959.
[59] W. E. Proebster and H. J. Oguey, "High-speed magnetic-film logic," *Digest of Tech. Papers*, 1960 Solid-State Circuits Conf., Philadelphia, Pa., pp. 22–23; February 10–12, 1960.
[60] D. W. Moore, "Evaporated films and digital computers," 1959 IRE WESCON Convention Record, pt. 4, pp. 32–39.
[61] K. D. Broadbent and F. J. McClung, "A thin magnetic-film shift register," *Digest of Tech. Papers*, 1960 Solid-State Circuits Conf., Philadelphia, Pa., pp. 24–25; February 10–12, 1960.

A Bibliographical Sketch of All-Magnetic Logic Schemes*

D. R. BENNION†, member, ire, H. D. CRANE†, senior member, ire, and D. C. ENGELBART†, member, ire

Summary—**An all-magnetic logic scheme is one with which a workable digital system could be constructed involving only magnetic elements, current-carrying conductors, and sources of clock pulses. Historical developments of both resistance schemes (dependent upon coupling-loop resistance) and nonresistance schemes (possessing at least first-order independence of coupling-loop resistance) are described, with reference to all relevant published work known to the authors. Included are: 1) schemes using electric-circuit transfer linkage with simple cores, multipath cores, and thin-film elements, and 2) schemes using continuous magnetic structures where transfer linkage is purely magnetic.**

Introduction

EVER SINCE magnetic elements were so elegantly adapted to computer memory use, serious efforts have been made to extend their use to logic systems as well . Motivation for such work lay primarily in the extremely high reliability and low cost of the magnetic devices themselves. Although advances in this field did not come simply, there still seemed to be a feeling that there was "something right" about magnetic logic. The primary difficulty was the inherent "bilateral nature" of a simple magnetic toroid. To overcome this difficulty, many efforts resorted to the use of diodes to achieve unilateral information flow, as reviewed by Haynes.[1] In many such circuits, the magnetic elements were used as (magnetic) amplifiers—operating in the half-cycle Ramey mode—and the logic was performed by the diodes. Although a number of clever computer systems resulted from these efforts, there was still considerable interest in an all-magnetic system. (It is usually agreed that by this term is meant a system composed of a magnetic elements and current conductors only—aside from the sources of clock power.)

The search for an all-magnetic logic was not motivated solely on the basis of an intellectual challenge, but by the desire to achieve a more homogeneous system. The use of diodes certainly negates much of the advantage of the magnetic elements—that is, low cost and essentially perfect reliability. Not only do the diodes represent an extra expense in themselves, but, also, their presence in the coupling loops results in still other disadvantages, such as the requirement of relatively large numbers of turns in the windings.

* Received by the PGEC, October 31, 1960; revised manuscript received, February 16, 1961.
† Computer Techniques Lab., Stanford Research Inst., Menlo Park, Calif.

[1] J. L. Haynes, "Logic circuits using square-loop magnetic devices: a survey," this issue, pp. 191–203.

Haynes[1] and Morgan[2] have provided very useful bibliographies on literature pertinent to digital magnetic techniques. Neither of these, however, is exhaustive relative to all-magnetic logic schemes. Because of the shortage of literature in this specific area, it was believed that a collection of all relevant published items with annotations, as well as comments on known unpublished work, would help provide both interested observers and active participants with increased perspective in the field.

Any comments on the workability of schemes will be concerned primarily with the question of whether bistable flux transfer can be achieved in a shift register of more than one bit. This is a sufficient condition for shift-register operation and a necessary condition for even the simplest (closed-loop) logic systems other than shift registers.

"Receiver loading" is a term that is used here to mean loading on the receiving core, during transfer, of such a degree that it is impossible to achieve sufficient flux gain regardless of the coupling-loop turns ratio. "Back transfer" is used here to mean back coupling through an input loop of such an amount that bistable transfer of the following information bit—if not actually prevented—will be possible only within narrow drive tolerances. If there is a back-transfer problem, a single-bit closed-loop register may work well, but a multibit register would work only with significantly reduced tolerances, if at all.

Techniques for flux transfer are divided into two main types: resistance schemes and nonresistance schemes. A resistance scheme is one in which, at certain times in each clock cycle, flux may be reversed in one or more paths linking a coupling loop without "opposition" from flux reversal in some other path or paths linking the loop. All other transfer schemes are of the nonresistance type; these are divided into two subtypes: 1) those using coupling loops (with flux "opposition" always occurring in each loop), and 2) those in which conductive coupling loops are replaced by direct magnetic coupling between stages. The term "continuous structures" is sometimes used to refer to the complex multipath cores required for the latter type of transfer.

The question of whether or not the conductive coupling loops between logic stages can be eliminated has always been an enticing one. There is no question that a combination scheme depending on magnetic coupling between some stages and conductive coupling between others could be devised, since the required flux-gain properties could be obtained by use of a limited number of coupling loops with turns ratio greater than unity. The crucial question has been whether bistable flux-gain properties can be obtained by use of magnetic paths only, in which case no coupling loops at all would be required. If so, then a complete logic system could in principle be realized in a single complex multipath core. Items [50]–[53] represent the published material along this line to date.

In illustration of the distinction between resistance and nonresistance schemes, consider the case of Magnetic Paths 1 and 2 linking a coupling loop of resistance R_l and inductance L_l. Then at every instant (given appropriate definitions of polarities),

$$N_1\frac{d\phi_1}{dt} + N_2\frac{d\phi_2}{dt} + R_l i_l + L_l\frac{di_l}{dt} = 0. \tag{1}$$

Assume that the loop current i_l is initially zero. After flux transfer, from Path 1 to Path 2, and after i_l has again returned to zero, we can write

$$N_1\Delta\phi_1 + N_2\Delta\phi_2 + R_l\int i_l dt = 0. \tag{2}$$

For a resistance scheme, it will be true in some cases that $\Delta\phi_2 = 0$ when $\Delta\phi_1 \neq 0$ (during a "priming" phase of operation). Therefore, the flux linkages $N_1\Delta\phi_1$ must be balanced by the term $R_l/i_l dt$ (sometimes referred to as the "flux dissipation in resistance"). In fact, from (2),

$$\left|\int i_l dt\right| = N_1\left|\Delta\phi_1\right|/R_l, \tag{3}$$

which shows that the electric charge which must circulate in the coupling loop, in order to "dissipate" a given $N_1\Delta\phi_1$, is inversely proportional to R_l. Hence, since it can be shown that the magnitude of i_l is always limited during such a "priming" operation, the time required for this operation will become very large if R_l is made very small. On the other hand, during a transfer phase of operation, (2) may be rewritten to yield the following expression for coupling-loop flux gain:

$$\text{gain} = \left|\frac{\Delta\phi_2}{\Delta\phi_1}\right| = \frac{N_1}{N_2} - \frac{R_l}{N_2}\left|\frac{\int i_l dt}{\Delta\phi_1}\right|. \tag{4}$$

Hence R_l cannot be very large, or it will be impossible to achieve sufficient flux gain. When actual numerical values are considered, it turns out that a "priming" phase of operation always occurs relatively slowly. For a nonresistance scheme, where flux switching in one leg would always be opposed by switching in the other leg, there is no "priming" phase of operation, and hence choice of R_l is governed by transfer-phase considerations only. The over-all result is that the maximum inherent speed capability of nonresistance schemes is usually significantly greater than that of resistance schemes (as is demonstrated for some specific schemes in [19]). In exchange, significantly greater tolerances on drive currents are usually obtainable for resistance schemes (as is demonstrated for a specific case in [20]).

Some of the major advances in the development of re-

[2] W. S. Morgan, "Bibliography of digital magnetic circuits and materials," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-8, pp. 148–158; June, 1959.

sistance schemes have been very recent, although the earliest schemes were of these type. As far as publication is concerned, it appears that after early efforts on resistance schemes, there was a period of concentration on nonresistance schemes, then a recent period with several papers on resistance schemes again. It is believed that both types have many possibilities and that substantial additional research could profitably be applied to both.

If there are any significant results in the field of all-magnetic logic that are not covered by items in this bibliography, we urge those who know of them to step forth and speak.

Bibliography on Resistance Schemes

[1] A. Wang, "Investigations for Design of Digital Calculating Machinery," Harvard Computation Lab., Harvard University, Cambridge, Mass., Progress Rept. No. 2, sect. IV, pp. 6–18; September, 1948.

Controlled nonlinear inductances are used in coupling loops in order to prevent both receiver loading and back transfer. Bistable operation of a closed-loop one-bit shift register was obtained; it is therefore apparent that receiver loading was prevented. With the scheme as given, however, it does not appear likely that back transfer as well as receiver loading could be prevented well enough to result in a workable multibit register.

[2] T. J. Rey, "Registers such as are Employed in Digital Computing Apparatus," U. S. Patent No. 2,683,819; July 13, 1954 (filed May 28, 1952).

The diodes in a one-diode-per-loop shift register are replaced by biased saturable inductors. The technique as shown prevents back transfer, but it would not at the same time prevent receiver loading; hence it would be impossible for sufficient flux gain to be achieved, and the scheme therefore appears unworkable.

[3] G. R. Briggs, "A Magnetic Core Gate and its Application in a Stepping Register," M.I.T., Cambridge, Mass., Digital Computer Lab. Engrg. Note E-475; October, 1952.

Two square-loop cores, A and B, are wired into each coupling loop between storage cores. When transfer is occurring through a given loop, the two cores are held in a saturated state by an auxiliary current. Receiver loading and back transfer are inhibited, respectively, by the switching of Core A in the succeeding loop and of Core B in the preceding loop. The result is a workable scheme requiring six cores per bit.

[4] R. C. Minnick, "The Use of Magnetic Cores as Switching Devices," Ph.D. dissertation, Div. Appl. Sci., Harvard University, Cambridge, Mass.; 1953.

[5] R. C. Minnick, "Magnetic switching circuits," *J. Appl. Phys.*, vol. 25, pp. 479–485; April, 1954.

A means for performing logic in diodeless coupling loops is presented in the above two items. However, no means is given for stringing logic stages together without the use of diodes in alternate coupling loops. Hence the scheme is not an all-magnetic one (although it does appear to require a much lower ratio of diodes to cores than previous core-diode schemes).

[6] N. B. Saunders, "Magnetic Memory Circuits Employing Biased Magnetic Binary Cores," U. S. Patent No. 2,781,503; February 12, 1957 (filed April 29, 1953).

A single biased core is used in each coupling loop with the aim of preventing back transfer, but it does not appear as though the method, as given, could achieve its objective. The idea of a bi-directional register using a turns ratio of unity is expressed, but the scheme provides no means for achieving sufficient gain in this case.

[7] L. A. Russell, "Diodeless magnetic core logical circuits," 1957 IRE National Convention Record, pt. 4, pp. 106–114.

Two different schemes employing four cores per bit are described. The "Type-I" scheme is similar to that of Saunders [6] and, like it, does not prevent back transfer.[3] The "Type-II" scheme interchanges the roles of the cores in a way that solves the back-transfer problem while still preventing receiver loading, resulting in an improved four-core-per-bit scheme.

[8] M. K. Haynes, L. A. Russell, J. J. Coughlin, W. A. Crapo, J. A. Kauffmann, N. F. Lockhart, R. F. Rutz, R. M. Tomasulo, and J. N. Cole, "Improvement Program for Magnetic Logical Circuits, Physical Research Effort," IBM Corp. Res. Lab., Poughkeepsie, N. Y., Terminal Rept.; June 1, 1956 to May 15, 1957.

Additional details are given on the schemes described in the previous item.

[9] G. R. Briggs and A. W. Lo, U. S. Patent No. 2,968,795; January 17, 1961 (filed May 1, 1957).

The concept of a resistance scheme using multipath cores is introduced. Flux is "primed" around a minor aperture in order to avoid back transfer and to aid the achievement of transfer without receiver loading.

[10] L. S. Onyshkevych, "Analysis of Circuits with Multiple-Hole Magnetic Cores," M.I.T. Res. Lab. of Electronics, Cambridge, Mass., Tech. Rept. 329; July 9, 1957.

Additional material on resistance schemes using multipath cores is presented. The treatment is brief but includes several ideas common to such schemes developed independently at a later date.

[11] "Magnetic Logic Core Improvement Program, Final Report," IBM Corp., Kingston, N. Y., Contract No. AF 30(635)-3130; January 1, 1958.

Diagrams of some resistance schemes are shown, but the principles of operation are not described.

[12] D. C. Engelbart, "High Speed Components for Digital Computers," Stanford Res. Inst., Menlo Park, Calif., Quarterly Reports 3-Pt. A, and 4-Pt. A, Contract AF 33(616)-5804; August, 1959.

[13] W. E. Proebster, S. Methfessel, and C. O. Kinberg, "Thin Magnetic Films," presented at the International Conf. on Information Processing, Paris, France; June 15–20, 1959.

[14] M. W. Green, "High Speed Components for Digital Computers," Stanford Res. Inst., Menlo Park, Calif., Final Engrg. Rept.-Pt. A, Contract AF 33(616)-5804; December, 1959.

[15] W. E. Proebster and H. J. Oguey, "Thin magnetic films for logic and memory," *Digest of Tech. Papers*, 1960 Solid-State Circuits Conference, Philadelphia, Pa., February 10–12, 1960; pp. 22–23.

In the four items above, resistance schemes making use of the coherent rotational switching properties of thin films are described.

[16] "Final Report on the Development and Use of Multiaperture-Core Flux Logic Devices to Perform Logical Functions in Digital Data Processing," IBM Corp. Federal Systems Div., Owego, N. Y., vol. II, Appendix A, Contract No. AF 33(600)-31315; December, 1959.

[17] U. F. Gianola, "Integrated magnetic circuits for synchronous sequential logic machines," *Bell Sys. Tech. J.*, vol. 39, pp. 295–332; March, 1960.

[18] H. P. Zeiger, "Diodeless Magnetic-Core Logic Circuits," M.S. thesis, M.I.T., Cambridge, Mass.; June, 1960.

[19] D. R. Bennion, "A note on magnetic shift registers," IRE Trans. on Electronic Computers, vol. EC-9, p. 262; June, 1960.

[20] D. R. Bennion, "MAD-resistance type magnetic shift registers," *Proc. Special Technical Conf. on Nonlinear Magnets and Magnetic Amplifiers*, Philadelphia, Pa., October 26–28, 1960, AIEE, New York, N. Y., pp. 96–112; 1960.

[21] D. Nitzan, "Analysis of MAD-R shift register and driver," *Proc. Special Technical Conf. on Nonlinear Magnets and Magnetic Amplifiers*, Philadelphia, Pa., October 26–28, 1960, AIEE, New York, N. Y., pp. 113–133; 1960.

[22] AMP Inc., Bulletins #610A, 610B, 610C, Harrisburg, Pa.; 1960.

[23] U. F. Gianola, "The Possibilities of All-Magnetic Logic Circuitry," presented at Conf. on Magnetism and Magnetic Materials, New York, N. Y.; November 14–17, 1960. *J. Appl. Phys.*, vol. 32, suppl., pp. 27S–34S; March, 1961.

The above group, beginning with [16], provides many details on resistance schemes employing multipath cores. Most of this material is concerned with a scheme employing a "holding" mmf to prevent back transfer.

[24] R. H. Tancrell, "Impulse Selection for Core Logic," presented at Conf. on Magnetism and Magnetic Materials, New York, N. Y.; November 14–17, 1960. *J. Appl. Phys.*, vol. 32, suppl., pp. 40S–41S; March, 1961.[4]

During *one* transfer, a spike of coupling-loop current (in a single-turn loop) is employed to reduce the dc threshold of the receiver rather than to transfer flux with unity gain. A drive current of amplitude lower than the normal dc threshold can then switch the receiver, resulting in sufficient over-all flux gain (only possible with certain materials). This result has been achieved with a register configuration for which sufficient gain by means of coupling-loop turns ratio only would be prevented by receiver loading. The gain mechanism used is similar to the ϕ^* flux gain reported in [27]–[29],

[3] In recent personal communication, Russell has assured us that multibit Type-I registers have been operated with adequate drive-current tolerances at low speed. This result suggests that we may have been treating the back-transfer problem too harshly.

[4] Tancrell has recently informed us that material on this scheme is also available in Rept. #52G-0014, a Lincoln Lab. internal group report.

but the author reports means for obtaining a temporary reduction (of order of a microsecond or less) as an alternative to a permanent reduction of threshold.

Bibliography on Nonresistance Schemes

In the case of resistance schemes, the early use of single-path cores gave way to substantial use of multipath cores later. In contrast, most of the early work with nonresistance schemes (except for the second item listed below) was based on the use of multipath cores.

[25] H. D. Crane, "Multi-Aperture Magnetic Devices for Computer Systems," Special Rept. to Burroughs Corp. (Paoli, Pa.), Stanford Res. Inst., Menlo Park, Calif.; February, 1957.

[26] M. K. Haynes, *et al.* (see [8] for complete data), sect. 1, pp. 1–6.

The "high speed diodeless transfer circuit" described in this report employs a three-phase clock cycle and three coupling loops per bit, and it requires six single-path cores per bit. During propagation of a *one*, a single "relief" core in a given loop: 1) sets to prevent receiver loading during transfer through the preceding loop, 2) stays set (with the help of positive bias) during transfer through the given loop, and 3) is reset to prevent back transfer during transfer through the succeeding loop.

[27] D. R. Bennion, H. D. Crane, and F. Heinzmann, "Multi-Aperture Magnetic Devices for Computer Systems," Tech. Rept. 2 to Burroughs Corp. (Paoli, Pa.), Stanford Res. Inst., Menlo Park, Calif.; October, 1957.

[28] D. R. Bennion, H. D. Crane, and F. Heinzmann, "Multi-Aperture Magnetic Devices for Computer Systems," Final Rept. to Burroughs Corp. (Paoli, Pa.), Stanford Res. Inst., Menlo Park, Calif.; January, 1958.

[29] H. D. Crane, "A high speed logic system using magnetic elements and connecting wire only" *Proc. 1958 Conf. on Non-Linear Magnetics and Magnetic Amplifiers*, Los Angeles, Calif., August 6–8, 1958; AIEE, New York, N. Y., pp. 465–482; August, 1957.

In [25], [27]–[29], a workable scheme employing multipath cores (multi-aperture devices, or MADs) is described. The minor aperture properties of the cores are used to prevent back transfer and receiver loading. Methods for achieving general logic are also indicated.

In [27]–[29], operable registers with unity-turns-ratio coupling loops are described. Because complete symmetry of wiring is possible, these registers can be bidirectional. Internal gain is achieved by the so-called ϕ^* flux-gain mechanism. (Since with unity turns ratio, the coupling loops do not contribute to flux gain—but only to losses—the positive results obtained suggest the possibility of eliminating the coupling loops and obtaining analogous register operation in a continuous multiaperture strip of material. Efforts to realize this possibility in the laboratory have thus far not been successful.)

[30] N. S. Prywes, "Diodeless magnetic shift registers utilizing transfluxors," IRE Trans. on Electronic Computers, vol. EC-7, pp. 316–324; December, 1958.

Registers are described which employ the same basic scheme of transfer underlying the work reported in items [25], [27]–[29]. The idea of eliminating coupling loops to obtain a "continuous-structure" register is expressed, but no means is given for solving the gain problem. (Another scheme that leaves the gain problem unsolved is found in [17].)

[31] H. D. Crane, "A high-speed logic system using magnetic elements and connecting wire only," Proc. IRE, vol. 47, pp. 63–73; January, 1959.

This paper is a revised version of [29]. (The basic scheme used in items [25], [27]–[31] is also touched on in [16].)

[32] E. P. Stabler, "Square-Loop Magnetic Logic Circuits," Electronics Lab., General Electric Corp., Syracuse, N. Y., Tech. Information Ser. R59ELS8; January 16, 1959.

[33] D. C. Engelbart, "A new all-magnetic logic system using simple cores," *Digest of Tech. Papers*, 1959 Solid-State Circuits Conference, Philadelphia, Pa., February 12–13, 1959; pp. 66–67.

[34] D. C. Engelbart, "High Speed Components for Digital Computers," Stanford Res. Inst., Menlo Park, Calif., Quarterly Rept. 2, Contract AF 33(616)-5804; February, 1959.

[35] E. P. Stabler, "Square-loop magnetic logic circuits," *Proc. Western Joint Computer Conf.*, San Francisco, Calif., March 3–5, 1959; pp. 47–53.

In the above four items, it is shown how multipath cores used in previous schemes can be replaced by arrays of single-path cores.

In [32] and [35], there is also some suggestion of means for eliminating coupling loops in circuits to yield magnetically coupled circuits, but no solution is given for the gain problem.

[36] D. R. Bennion and H. D. Crane, "Design and analysis of MAD transfer circuitry," *Proc. Western Joint Computer Conf.*, San Francisco, Calif., March 3–5, 1959; pp. 21–36.

Additional details and variations are given on the transfer circuitry for the techniques described in items [25], [26]–[29], [31].

[37] D. C. Engelbart, U. S. Patent Application No. 847,149 (filed October 19, 1959).

A means is described for using single-path and multipath cores together in a "hybrid" circuit in order to obtain some of the advantages of both in common.

[38] H. D. Crane, "Research on General Digital Logic System Utilizing Magnetic Elements and Wire Only," AF Cambridge Res. Ctr., Bedford, Mass., Quarterly Status Rept. 1, Contract No. AFCRC-19(604)-5909; September, 1959.

[39] E. K. Van De Riet and C. H. Heckler, Jr., "Research on General Digital Logic Systems Utilizing Magnetic Elements and Wire Only," AF Cambridge Res. Ctr., Bedford, Mass. Final Engrg. Rept., Contract AFCRC-19(604)-5909; October, 1960.

[40] H. D. Crane and E. K. Van De Riet, "Circuit approach for an all-magnetic computing system," *Digest of Tech. Papers*, 1961 Solid-State Circuits Conf., Philadelphia, Pa.; February, 1961.

[41] H. D. Crane and E. K. Van De Riet, "Design of an all-magnetic computing system, Part I—circuit design," this issue, pp. 207–220.

[42] H. D. Crane, "Design of an all-magnetic computing system, Part II—logical design," this issue, pp. 221–232.

In the above five items, the design, construction, and operation of an all-magnetic decimal arithmetic unit based on the use of the hybrid technique of item [37], with elaborations, is described.

[43] H. D. Crane, "Magnetic Core Logic Element," U. S. Patent No. 2,935,622; May 3, 1960 (filed June 12, 1958).

[44] D. R. Bennion, "A new multiaperture magnetic logic element," *J. Appl. Phys.*, vol. 31, pp. 129S–130S; May, 1960.

[45] D. R. Bennion, "Research on Multiaperture Magnetic Logic Devices," Stanford Res. Inst., Menlo Park, Calif., Tech. Rept. 1, Contract Nonr. 2712(00); May, 1960.

In the above three items, a class of multipath cores that can be wired for either logical positive or logical negative transfer is described. With cores of this type, logic capability can be achieved in circuits containing cores of a single mold only.

[46] H. D. Crane, "Sequence detection using all-magnetic circuits," IRE Trans. on Electronic Computers, vol. EC-9, pp. 155–160; June, 1960.

A particularly simple application of all-magnetic arrays to sequence detection is described.

[47] S. B. Yochelson, "Diodeless Core Logic Circuits," Goodyear Aircraft Corp., Akron, Ohio, GER-9798; June 15, 1960.

[48] S. B. Yochelson, "Diodeless core logic circuits," 1960 IRE WESCON Convention Record, pt. 4, pp. 82–95.

[49] D. R. Bennion, "Diodeless core logic circuits—S. B. Yochelson," review R61-27, IRE Trans. on Electronic Computers, vol. EC-10, p. 114; March, 1961.

The above three items deal with a scheme that employs a four-phase clock cycle and single-path cores, but which uses a different mode of transfer than the scheme described in items [33] and [34].

[50] A. H. Bobeck and R. F. Fischer, "Reversible, Diodeless, Twistor Shift Register," 1958 Conf. on Magnetism and Magnetic Materials, Philadelphia, Pa., November 17–20, 1958; *J. Appl. Phys.*, vol. 30, pp. 43S–44S; April, 1959.

[51] D. W. Moore, "Magnetic domain switching in evaporated magnetic films," *Proc. 1959 Electronic Components Conference*, May 6–8, 1959, Philadelphia, Pa., AIEE, New York, N. Y., pp. 11–14.

[52] K. D. Broadbent and F. J. McClung, "Thin magnetic films for logic and memory," *Digest of Tech. Papers*, 1960 International Solid-State Circuits Conf., Philadelphia, Pa., February, 1960; pp. 24–25.

[53] K. D. Broadbent, "A thin magnetic film shift register," IRE Trans. on Electronic Computers, vol. EC-9, pp. 321–323; September, 1960.

The above four items describe magnetically coupled schemes of the type involving moving domains in thin continuous strips of metallic magnetic material. (This type of scheme is also discussed in [17].)

Although still low, the number of people working on all-magnetic logic seems to have increased considerably during the past year or two. We end this sketch with the hope that this trend will continue and that this end will be just a beginning.

Design of an All-Magnetic Computing System: Part I—Circuit Design*

H. D. CRANE†, SENIOR MEMBER, IRE, AND E. K. VAN DE RIET†, MEMBER, IRE

Summary—**This paper describes the circuits used in a decimal arithmetic unit which utilizes ferrite magnetic elements and copper conductors only. The arithmetic operations of addition, subtraction, and multiplication are performed with a product and sum capacity of three decimal digits. The sole logical building block of this system is a two-input inclusive-OR module with a fan-out capability of three with any desired logical positive and negative combination. The system involves the use of some 325 modules, each of which contains two magnetic multiaperture devices (MAD's). This paper gives a complete description of the circuit and physical arrangement of the machine. The system is controlled from a manual keyboard, and readout from the machine is via incandescent lamps controlled directly from the MAD elements, no intermediate elements being required.**

The "worst case" drive-pulse amplitude range for the completed machine, varying all clock pulses simultaneously, is ±10 per cent.

I. Introduction

THE DESIGN of a decimal arithmetic unit constructed entirely with all-magnetic circuits is described. Primary attention is paid in Part I to the particular forms of the magnetic circuits used. (Logical synthesis techniques and the system design are discussed in Part II.)

As pointed out in an accompanying paper,[1] the search for an all-magnetic logic system has kept a number of people occupied over the past few years. Although logically operable systems had been previously conceived, until now only very small pieces of such systems have actually been constructed, *e.g.*, small shift-register structures and the like. One of the primary purposes for building this particular computing structure, therefore, was to try to determine how a relatively complex all-magnetic logic system could be expected to perform. All-magnetic circuits, at least those thus far conceived, are very difficult to analyze. However, sufficient empirical feel had been accumulated to warrant building a relatively large system to discover how valid our intuitive predictions "in the small" would be "in the large." For example, the two most common performance complaints against all-magnetic systems are 1) their low gain (*i.e.*, fan-out capability), and 2) the small per cent tolerances on the drivers. To test the legitimacy of these complaints, we designed a circuit that would theoretically give a reasonable fan-out and reasonable drive tolerance. That these fan-outs and tolerances could actually be achieved in small circuit arrays (such as registers) was easily verified. The real question, however, was whether or not, in view of the lack of complete analytic tools, we could reasonably expect the performance to hold up in a relatively large system, or whether some unpredictable factor, or factors, would militate against it? And if so, what were they? As it turned out, the empirical design method held up very well; the performance is essentially as predicted. Another pair of complaints commonly leveled against all-magnetic logic (and, as a matter of fact, against diode-magnetic logic systems as well) is 3) the complexity of wiring, and 4) the large clock power required. This problem of wiring created no small concern. As it turned out, the wiring scheme that was used (see Section VI) led to no particular difficulties. However, its practical expandability to still larger systems requires further study. (The machine that was built is sufficiently large so that its performance data are significant. However, it is not so large that it can be considered a highly interconnected system, topologically.) As to the large clock power required, we agreed to ignore this aspect in the beginning, since it seemed more important to get a feel for the possibility of designing an operable all-magnetic logic system. Thus, the initial contention was to design a sound logic system, and supply it with whatever power sources were required. The completed system is easily driven with ordinary laboratory power pulsers; even in this area there seems to be a considerable amount of simplicity that can be achieved.

II. Boundary Conditions

The main schemes available from which to choose all-magnetic logic circuitry are 1) those in which one logically relies on resistance in the coupling loop, and 2) those nonresistance schemes in which resistance in the loop is not logically required.[1] Although resistance schemes lead to operating speeds that are lower by several factors, circuits using such schemes generally exhibit much greater operating tolerances. Primary reasons for choosing a nonresistance system were: 1) at the time that the program was started, insufficient experience had been accumulated with resistance schemes, and 2) it was felt that if we succeeded in building a sys-

* Received by the PGEC, October 31, 1960; revised manuscript received, January 18, 1961. The work reported here was carried on at Stanford Research Institute, Menlo Park, Calif., under the sponsorship of the AF Cambridge Research Labs., Bedford, Mass., Contract No. AF 19(604)-5909.

† Stanford Research Institute, Menlo Park, Calif.

[1] D. R. Bennion, H. D. Crane, and D. C. Engelbart, "A bibliographical sketch of all-magnetic logic techniques," this issue, pp. 203–206.

tem based on resistance circuits, we would still not be sure that the higher-speed, basically lower-tolerance circuit approaches would be operable. On the other hand, now that we have successfully realized a system using a nonresistance scheme, we have little doubt that systems involving resistance schemes could be made to operate as well. In fact, we feel that the successful operation of this system represents a significant step toward establishing the practical usefulness of all-magnetic logic systems.

Thus, the choice was to proceed with a resistanceless logic system. At this point, there existed two possibilities: 1) circuits using multiaperture elements (MAD's), or 2) circuits based on Engelbart's all-core equivalents,[2] in which only simple toroids are involved. Actually a hybrid scheme,[3] using some of both approaches, was adopted, but for two reasons the MAD scheme was accented. Very little actual experience had been obtained with circuits of the Engelbart type. Further, what little had been obtained indicated that the analysis of these circuits might be considerably more difficult than for MAD circuits; at a minimum, additional work would be required to establish even empirical design procedures.

Thus, MAD's were to be used primarily. At the beginning of the program, two types of molded MAD's (each having four minor apertures) were available in relatively large numbers. However, these elements differed considerably in their dimensions. The element chosen for use was the physically larger of the two. The choice was based on the fact that the larger element possessed a more favorable geometry as far as expected circuit performance is concerned. (The dimensions of the selected element are illustrated in Fig. 1.) The all-core

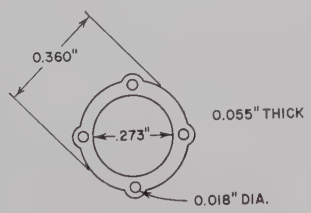


Fig. 1—Basic MAD element. (Flux capacity = 0.56 volt-μsec.)

type of circuitry was included to the extent of achieving logical negation. Although logical negation can be achieved in MAD circuits, elements having special geometrical design are required, and no such elements were available at the time. Hence, in the system to be described, negation is obtained by use of ordinary toroids. (The details are covered later in the paper.)

Finally, all decisions made in the development of the final circuit form were based almost completely on a desire to obtain optimum circuit performance, to increase the probability of successful over-all operation. Performance, in terms of driver tolerance, was therefore the primary consideration. Maximizing speed, or minimizing element count, for example, were not deemed appropriate objectives for this "feasibility" study. However, now that the first system is completed, we have no doubt that the smaller elements would have also worked satisfactorily, although with some degradation of performance.

III. Basic MAD System

The operational and design features of basic MAD circuits have been covered in other publications.[4,5] One of their most important features is that binary data transfer between elements is achieved over coupling loops containing only conductive wire. In this system, a shift-register takes the form indicated in Fig. 2, where

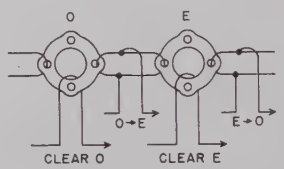


Fig. 2—Basic MAD shift register.

alternate elements have been labeled O (Odd) and E (Even), and the driver consists of four clocks: two Advances (Adv $O \rightarrow E$ and Adv $E \rightarrow O$) and two Clears (Clear O and Clear E). The transfer process is nondestructive in the transmitter so that the transmitting element must be explicitly cleared before being read into again. Thus, the basic clock cycle is

$$\cdots \text{Adv } E \rightarrow O, \quad \text{Clear } E, \quad \text{Adv } O \rightarrow E, \quad \text{Clear } O, \cdots .$$

Although the basic clock system involves four clocks, logically this is a two-clock system. Thus, in any two-clock logic system, there are two groups of storage elements, the O's and the E's. At one clock time, logical combinations of, for instance, the states stored in the O elements are transmitted to the E's. At the next clock time, logical combinations of the E states are transmitted to the O elements, and so on. In this case, specific clear pulses are required between clocks only because the transmission from one set to the other is nondestructive. Although E and O notations have been used extensively in the past to denote names for these two sets of elements, it was more convenient in this program to change

[2] D. C. Engelbart, "A new all-magnetic logic system using simple cores," presented at the 1959 Solid State Conf., University of Pennsylvania, Philadelphia; February 12–13, 1959. The paper is summarized in "The Conference Digest," pp. 68–69.

[3] D. C. Engelbart, "Combined synthetic and multiaperture magnetic core system," Patent Application No. 847149; October 19, 1959.

[4] H. D. Crane, "A high-speed logic system using magnetic elements and connecting wire only," Proc. IRE, vol. 47, pp. 63–73; January, 1959.

[5] D. R. Bennion and H. D. Crane, "Design and analysis of MAD transfer circuitry," *Proc. 1959 Western Joint Computer Conf.*, San Francisco, Calif., March 3–5, pp. 21–36.

the names to ○ (circle) for even and □ (square) for odd. This is so, since on logical schematics it is convenient to graphically indicate to which group any particular logic module belongs by simply drawing the module as a ○ or □ (even or odd). Thus, in this notation, the basic clock cycle is

$$\cdots \text{Adv} \bigcirc \rightarrow \square, \text{Clear} \bigcirc, \text{Adv} \square \rightarrow \bigcirc, \text{Clear} \square, \cdots.$$

IV. Circuit Development

The present system was designed on the basis of a standard two-input, three-output logic module.[6,7] Given two inputs x, y, the module forms the inclusive-OR $x+y$. Each output can be independently selected to be positive or negative. Thus, each output can be selected to be $x+y$, or $\overline{x+y}=\bar{x}\bar{y}$. The latter is the standard NOR function.

In the development that follows, primary emphasis is on the logical development of the module circuit. For simplicity, the details about particular bias and drive arrangements are minimized. At the completion of the present section (Section IV), exact details of the final circuit are presented.

It should be kept in mind throughout this development, that the design was based on a specific MAD element having four minor apertures. The design might have taken many other forms, had different elements been used.

A. Multiple Outputs (Fan-Out)

An elementary positive circuit is indicated in Fig. 3(a). This circuit is defined as positive (as opposed to negative) since the ○ element, following a □→○ pulse, assumes the same state as the □ element. Fig. 3(b) illustrates an arrangement for controlling three outputs from the same □ element, during the □→○ pulse. Although it has been experimentally demonstrated that a number of outputs can be simultaneously controlled in this fashion, the tolerances on the □→○ pulse became poorer as more loads were added. In line with the philosophy of maximizing the allowed clock pulse tolerances, the clock system of Fig. 3(c) was adopted to achieve the required fan-out of three. In this arrangement, each transfer loop connects to a separate transmitter aperture and is energized at a different clock time. Thus, during the $(\square\rightarrow\bigcirc)^1$ pulse (the superscript denotes the time sequence of the transfer pulses), the first transfer loop is excited. Recall that the transfer process is nondestructive to the transmitter, in the sense that (at least in first-order analysis) any flux switched in the transmitter as a result of the transfer process, switches only locally about the output aperture. Hence, the flux states about the other output apertures are substantially unaffected by the first transfer. At $(\square\rightarrow\bigcirc)^2$, the second transfer loop is excited, and at $(\square\rightarrow\bigcirc)^3$, the third loop is excited.

To achieve the required fan-out in the ○→□ direction, three other Advance pulses are used, so that the resulting clock system uses eight pulses with the following cycle arrangement:

$$\cdots (\square \rightarrow \bigcirc)^1, (\square \rightarrow \bigcirc)^2, (\square \rightarrow \bigcirc)^3, \text{Clear} \square,$$
$$(\bigcirc \rightarrow \square)^1, (\bigcirc \rightarrow \square)^2, (\bigcirc \rightarrow \square)^3, \text{Clear} \bigcirc, \cdots.$$

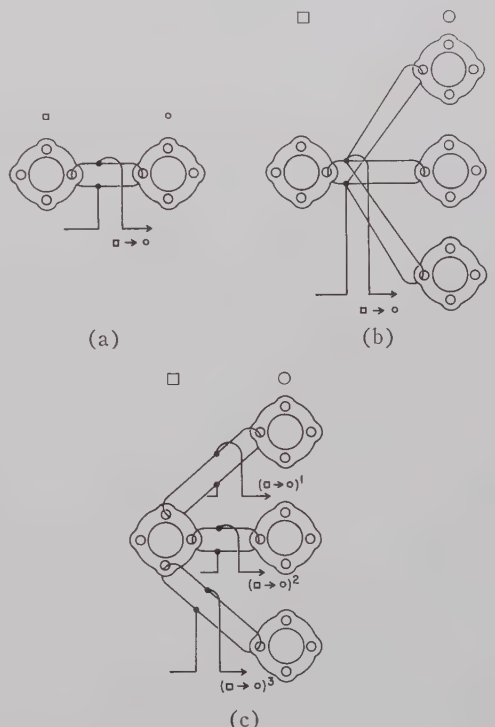


Fig. 3—Multiple output control.

The actual clock system used in the machine, however, consists of ten pulses. The need for the extra two clocks will be clarified at a later point in the discussion.

B. Flux Gain and Flux Clipping

In Fig. 4(a) a pair of elements connected in a closed loop are shown. This is equivalent to a one-bit shift register, and the circuit has the property of stably circulating a high level of flux over the coupling loops (one state) or a low level of flux (zero state).

In order for stable two-level operation to exist, the input (or received) flux ϕ_R during each transfer, and the input flux ϕ_R' during the previous transfer must be related as indicated in Fig. 4(b) and (c). The gain $G=\phi_R/\phi_R'$ is >1 in the interval $\phi_I<\phi_R'<\phi_U$ and is <1 in the interval $\phi_L<\phi_R'<\phi_I$, where ϕ_U is the (upper) *one*-level flux, and ϕ_L is the (lower) *zero*-level flux [Fig. 4(d)]. Thus, the transfer operation tends to increase a "low"

[6] H. D. Crane, "Research on the Realizability of an All-Magnetic Computing System," Stanford Res. Inst., Menlo Park, Calif., Rept. No. 1, Contract AF 19(604)-5909; October, 1959.

[7] E. K. Van De Riet and C. H. Heckler, Jr., "Research on General Digital Logic Systems Utilizing Magnetic Elements and Wire Only," Stanford Res. Inst., Menlo Park, Calif., Final Rept. Contract AF 19(604)-5909; October, 1960.

one level of flux toward ϕ_U, and to decrease a "high" *zero* level of flux toward ϕ_L. With such a gain relation, the operation stably protects against "*zero* build-up" and "*one* build-down."

With a coupling loop turns-ratio greater than unity (*i.e.*, $N_T > N_R$ where N_T is the number of turns on the transmitter and N_R the number of turns on the receiver), it is a straightforward matter to arrange a coupling loop to obtain the necessary relation between ϕ_R' and ϕ_R. Suppose that there are no losses in the coupling loop of Fig. 5(a). By integrating the relation $N_T \dot{\phi}_T = N_R \dot{\phi}_R$, which must hold at every instant of the switching period, we find the relation $N_T \phi_T = N_R \phi_R$ where ϕ_T and ϕ_R are the net flux changes in the transmitter and receiver. Then, provided $\phi_T = \phi_R'$,

$$G = \frac{\phi_R}{\phi_R'} = \frac{N_T}{N_R} . \tag{1}$$

This (ideal) relation between ϕ_R and ϕ_T is illustrated in Fig. 5(b), with $(N_T/N_R) > 1$. Notice that the linear relation holds only until ϕ_R saturates. Finally, in order to obtain the necessary relation of Fig. 4(c), consider the coupling loop of Fig. 5(c), in which a "clipper" core is added. This core is arranged to have a total flux linkage capacity ϕ_C, which is a relatively small fraction of the saturation flux linkage of the MAD's. However, the switching threshold of the clipper is much lower than that of the receiver, and therefore, its full capacity of flux will be substantially switched before the receiver begins switching at all. In this way, a constant amount of flux is subtracted from the quantity $(N_T/N_R)\phi_T$ when $> \phi_C$; if $(N_T/N_R)\phi_T < \phi_C$, then $\phi_R = 0$. The resulting relation between ϕ_T and ϕ_R is shown in Fig. 5(d), where the zero level of flux is shown ideally as $\phi_L = 0$. Notice that this curve has the proper form for bistable operation. If the flux clipper is cleared at the same time as the transmitter, none of the basic clock cycle operations is altered, and very good transfer loop operation is achieved. Effective flux "clipping" can be achieved by the flux losses in the inherent resistance and inductance of the coupling loop, but it is more advantageous to use explicit clippers, because the coupling loop impedance type of clipping requires relatively high impedance which is in conflict with the low impedance requirement of *one* transfer.

C. Flux Gain and Excess Current

In Fig. 6 a coupling loop is indicated with a transmitter-to-receiver turns ratio of n. For successful operation, certain current and flux relations must be satisfied simultaneously. Thus, in order to achieve flux gain, n must be greater than unity (assuming there are no other sources of flux gain, such as in the elements themselves). The higher n is, the higher is the flux gain; however, the higher the n, the poorer the current conditions become.

The maximum allowed value of Advance current I_A is determined by the *zero* transfer condition, while the minimum value is determined by the *one* transfer. For the circuit of Fig. 6, $NI_A \leq F_2$, where F_2 is the threshold mmf about the main aperture of the receiver. For $NI_A > F_2$, *zero* build-up tendencies enter strongly. With an Advance pulse magnitude $NI_A = F_2$, the receiver is essentially biased to threshold. If the transmitter is in the *one* state during the Advance pulse, adequate loop current (I_L) must flow if the receiver is to switch completely. The maximum value of loop current is F_2/n, assuming negligible magnetizing drop about the output aperture of the transmitter during switching. In order for the Advance pulse to have some allowed range, it must be possible to lower its value below its maximum and still have successful *one* transfer. Now, the lower n is,

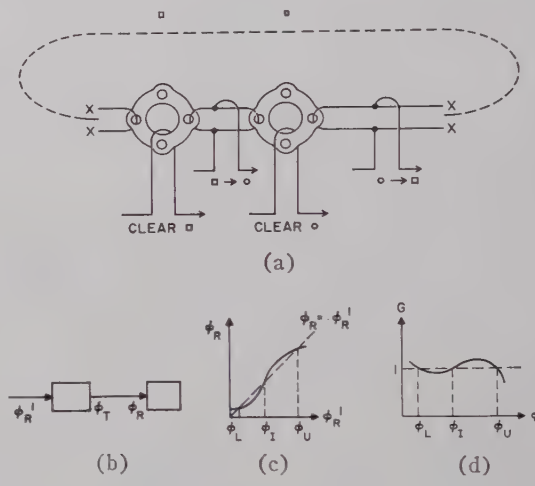


Fig. 4—One-bit MAD shift register and gain relations for bistable operation.

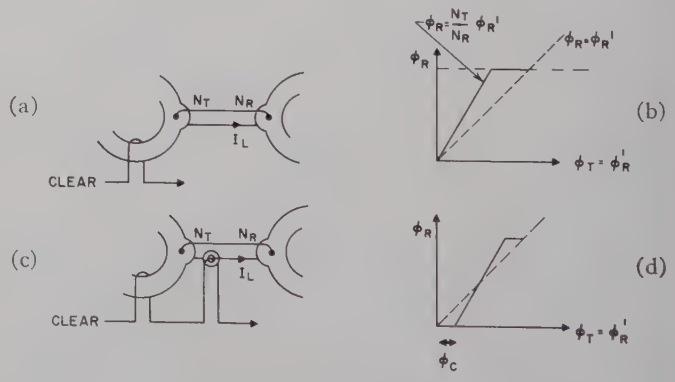


Fig. 5—Coupling loop arrangement for achieving proper gain relations.

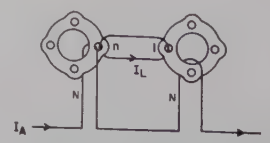


Fig. 6—Coupling loop with turns ratio of n.

the higher will be the loop current available to the receiver for any value of I_A, and hence, the lower the value I_A may become while sustaining proper *one* transfer. This results in a greater range.

The current available from the transmitter, via the coupling loop, is usually referred to as the "excess current." Clearly, the turns ratio n must be selected to satisfy the flux gain and excess current requirements simultaneously. Generally, values of n between 1 and 2 are common for these types of circuits. If a nonintegral ratio is desired, this can only be physically achieved by arranging N_R to be other than a single turn. Thus, for $n=1.2$, values $N_T=6$, $N_R=5$ would do. It may be noted that in circuits developed here, single-turn windings are used ($N_T=N_R=1$); actually, however, there are two transmitters per receiver, yielding an effective value of $n=2$. This arrangement is discussed in the following section (Section IV-D).

D. Gain-Excess Doubling

With presently available materials, it is difficult to design successfully a relatively large network with the regular circuit approach taken so far. To obtain increased design margins, a circuit is used which effectively doubles the gain-excess product,[8] so that for a given excess the gain is doubled, and vice versa.

Consider again the coupling loop of Fig. 6 with turns ratio n. Suppose that *one* and *zero* flux levels available at the transmitter end of the coupling loop are 10 units and 1 unit, respectively. Then a flux clipper of 1 unit would clear up the *zero* level, leaving 9 units of *one* flux. If two transmitters are connected in series, as in Fig. 7(a), each providing the same magnitude of flux, there would be 20 units and 2 units of flux available. With a 2-unit flux clipper there would result 18 units of *one* flux, or twice as much flux gain, but with no decrease in the amount of available excess current, *i.e.*, flux doubling. At the other extreme, if the two transmitters were connected with $n/2$ turns, as shown in Fig. 7(b), there would only be the original 10 units and 1 unit of flux available, but the excess current would be doubled, *i.e.*, excess-current doubling. For any particular value of transmitter turns between n and $n/2$, some combination of gain of flux and excess current is obtained.

For construction simplicity, there is strong motivation to use single-turn windings in the present system, and, in fact, the coupling loops are actually realized in the manner of Fig. 7(a), with $n=1$ (see Figs. 13, 14). This results in an arrangement which is much closer to flux doubling, in contrast to the other extreme of excess-current doubling since, were a single transmitter element to be used rather than a pair, the value of n would have been chosen close to unity.

By using two elements in this manner, in order to increase the gain-excess product, it then becomes necessary to consider means for insuring that these two elements are in the identical state before they are used as transmitters. One technique for accomplishing this is to transmit the same logic variable (labeled x) into each element during two Advance pulses as indicated in Fig. 8(a). An alternative technique is to transmit the vari-

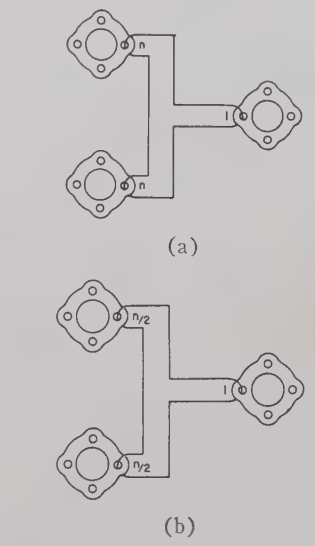


Fig. 7—Gain-excess-doubling circuit.

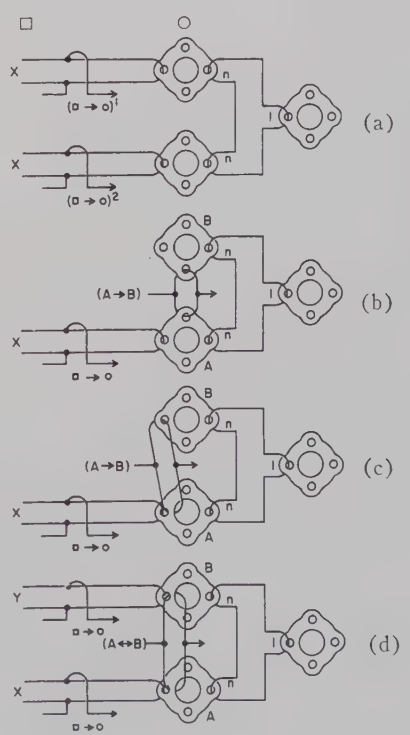


Fig. 8—Input to a gain-doubling module.

[8] D. R. Bennion, D. C. Engelbart, and E. K. Van De Riet, "Flux-doubling transfer between magnetic elements," Patent Application No. 34678; June 8, 1960.

able x to only a single element, A, and then to advance it over a normal coupling loop (labeled $A \rightarrow B$) to the other element of the pair, B, as in Fig. 8(b). This latter technique has the advantage that only a single □→○ pulse is required for transmitting the variable. The local coupling loop, $A \rightarrow B$, can be excited during the Clear □ pulse so that no extra clock time is required. If Element A receives a *one*, then during the Clear pulse, a loop current flows to switch flux locally about the input aperture of Core A. If the $A \rightarrow B$ loop is excited simultaneously, then the normal Advance drive for the $A \rightarrow B$ loop would have to be decreased because of the action at the input aperture. There is some advantage, therefore, to having the $A \rightarrow B$ coupling loop link the input aperture of Element A, as illustrated in Fig. 8(c). First, this reduces the number of apertures required; second, the mmf in the input winding, due to the Clear □ pulse, actually aids the $A \rightarrow B$ transfer. Note that any flux switched in Element A during the Adv □→○ pulse does not link the $A \rightarrow B$ loop.

By symmetry, it is clear that it is possible to bring a second input, labeled y in Fig. 8(d), to the B element. In this case, the loop previously labeled $A \rightarrow B$ should be relabeled $A \leftrightarrow B$, since the loop must transfer flux in either direction. Consider the x, y input combinations. If $x=y=0$, no flux is switched in the $A \leftrightarrow B$ loop during the $A \leftrightarrow B$ clock pulse. For $x=y=1$, equal amounts of flux are switched at both ends of the $A \leftrightarrow B$ loop, which is so connected that the induced voltages oppose, and again no loop current results. If either x or y, but not both, is a *one*, then during $A \leftrightarrow B$ pulse, a *one* will be transmitted to the "other" element so that both are *one*. Hence, both elements are always in the same state after the $A \leftrightarrow B$ pulse. Furthermore, this state is a *one* for all input conditions except $x=y=0$. Thus, this circuit forms the logic (Inclusive) OR function $(x+y)$. Note that the input loops may be excited simultaneously or not, since there is no coupling between the A and B elements during the entry phase.

By the nature of the symmetry, it is clear that the windings on either end of the $A \leftrightarrow B$ loop must be equal. Thus, no flux gain can be expected in this loop on the basis of turns ratio. (It should be noted that H. D. Crane[4] refers to a ϕ^* flux-gain mechanism which is independent of turns ratio, and by means of which flux gain greater than unity can actually be achieved even in coupling loops of unity turns ratio. It was further pointed out, however, that the magnitude of this effect is very material dependent, and also that it is difficult to obtain wide operating tolerance in circuits that depend entirely on this mechanism for flux gain. Therefore, no special attention was paid to this phenomenon in the choice of material for the elements of this system, and the proper performance of these circuits is not dependent upon it. Nevertheless, any flux gain obtained in the $A \leftrightarrow B$ loop in this manner is to our advantage.) For the case in which only one receiver element holds a *one*, and this *one* is transmitted to the other element, we might expect the received flux in the second element to be less than that initially received in the first element. Hence, we cannot expect to get exactly a doubling of the flux-excess product in this way, but, nevertheless, the improvement is significant. In order to get the most benefit, however, we want the least possible flux loss in this loop. Clearly, then, a clipper core is not used. Furtheremore, in the physical lay-out, the pair of elements of each module circuit are placed very close together, so that the flux losses in coupling loop impedances are very small, and moreover are the same in all modules. Thus, even with a unity turns ratio in this loop, the net flux gain will be only slightly less than unity. Thus, although the actual improvement in the gain-excess product may be just slightly less than the nominal factor of two, that in itself is a very large improvement.

E. The Clock Program

Although the $A \leftrightarrow B$ clock pulse of the previous section can logically occur simultaneously with the previous Clear clock pulse, it is indicated in this section why it was advantageous in this first machine design to actually separate them.

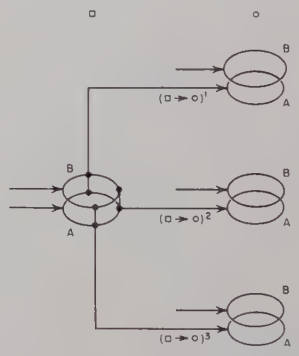


Fig. 9—Fan-out schematic diagram.

Assume that the $A \leftrightarrow B$ and Clear pulses do occur simultaneously. The schematic diagram of Fig. 9 represents a single □ element controlling the A inputs of three ○ modules. The B inputs of these modules connect to other sources. Assume that the □ element holds a *one*. Then, after all □→○ pulses, the Clear □ pulse returns the □ element to its *zero* state. Due to this switching, currents flow in the □→○ loops causing reversal of flux about the input apertures of the A elements. However, after the □→○ pulses, each of the corresponding B elements may be, independently, in a *one* or *zero* state. As a result of the simultaneous [Clear □, $(A \leftrightarrow B)$] pulse some $A \leftrightarrow B$ loops will be forced to carry current and others will not. The actual magnitude of current flowing in any particular □→○ loop, therefore, depends upon whether corresponding $A \leftrightarrow B$ loop current flows as well (*i.e.*, depends upon whether the corresponding B element holds a *one* or a *zero*). Hence, the currents in the □→○ loops during Clear are data-sensitive. For

example, for $B_1=B_2=B_3=1$, no currents flow in the $A\leftrightarrow B$ loops. For $B_1=B_2=B_3=0$, currents flow in all $A\leftrightarrow B$ loops with the result that the currents in the $\square\rightarrow\bigcirc$ loops are higher than for the previous case for the same strength of Clear $\square$. For mixed states still other conditions hold. With such pattern sensitivity possible, the circuit becomes difficult to design, since both strong transmitter clearing and reliable $A\leftrightarrow B$ transfer operation are required.

To overcome this possible cause of difficulty, the Clear and $A\leftrightarrow B$ transfer operations are separated. This adds two more pulse times to the clock program which now assumes the form

$$\cdots, (\square\rightarrow\bigcirc)^{1,2,3}, \quad (A\leftrightarrow B)_{\bigcirc}, \quad \text{Clear } \square,$$
$$(\bigcirc\rightarrow\square)^{1,2,3}, \quad (A\leftrightarrow B)_{\square}, \quad \text{Clear } \bigcirc, \cdots.$$

The notation $(\square\rightarrow\bigcirc)^{1,2,3}$ denotes the sequence $(\square\rightarrow\bigcirc)^1$, $(\square\rightarrow\bigcirc)^2$, $(\square\rightarrow\bigcirc)^3$. During the $(A\leftrightarrow B)_{\bigcirc}$ pulse (Fig. 9), the regions about the output apertures of the $\square$ transmitter switch locally as a result of currents in the $\square\rightarrow\bigcirc$ loops, but the $\square$ element is not cleared. During this operation, therefore, the transmitter element offers the same load to the $\bigcirc$ elements as three independent toroids, one in each loop. With this arrangement, the switching in each $A\leftrightarrow B$ loop is much less sensitive to the switching status in the other $A\leftrightarrow B$ loops.

After the $(A\leftrightarrow B)_{\bigcirc}$ pulse, the $\square$ elements are cleared. However, since the output apertures of the $\square$ element have already been switched during $(A\leftrightarrow B)_{\bigcirc}$, the Clear pulse can clear the $\square$ elements without causing any (substantial) currents in any coupling loops. Hence, there are no special requirements on the Clear pulse, and in fact a very strong Clear can be used, which is very desirable for reliable operation. Actually, the Clear will be used for one other simultanous operation to be defined below with respect to negation, but this will not affect the Clear operation as such.

F. Negative-Transfer (Synthetic-Aperture Technique)

The simple MAD elements of the previous discussions are capable only of logical positive transfer; negation, or negative transfer, is achieved in another manner adopted from a technique in which a simple MAD is replaced by an array of conventional magnetic toroids equal in number to the total number of apertures in the MAD, including one toroid for the main aperture. These toroids are referred to as synthetic apertures. In the full synthetic approach, separate toroids are used for both input and output apertures. For use here, however, only output aperture synthesis is utilized.

In Fig. 10 the output aperture of the MAD has been replaced by a separate toroid which is referred to as the output core. With the connection indicated in the figure, if the $\square$ element is in the *zero* state (as shown), the $\bigcirc$ element will receive a *one* during $\square\rightarrow\bigcirc$, and conversely. We briefly consider the operation. Assume that the $\square$ element is in the *zero* state. Then the elements have the states shown in the figure, *i.e.*, all are in their Clear state. However, the output core is in a state to switch with very low current during a $\square\rightarrow\bigcirc$ pulse, similar to the condition of a regularly set MAD transmitter. Hence, during a $\square\rightarrow\bigcirc$ pulse, the majority of the current flows in the receiver branch causing the receiver to set. Alternatively, assume that the $\square$ element had previously been set; during the setting, an amount of flux linkage is provided in the output loop which causes the output core to switch at a low-current level that does not affect the receiver. In this condition, during the $\square\rightarrow\bigcirc$ pulse, the transmitter branch has a high switching threshold, similar to a regularly cleared MAD transmitter, and a $\square\rightarrow\bigcirc$ pulse will not result in receiver switching. Thus a *zero* is transmitted, and the negative transfer function is demonstrated.

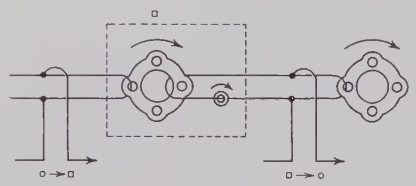


Fig. 10—Synthetic output aperture circuit.

More than a single negation output loop may be used. It may be noted that negative transfer is nondestructive to the transmitter in the same sense as is a positive MAD transfer. Hence, in cases of mixed positive and negative output there is no special concern for the order of read-out.

Nominally, the gain-excess relations are the same for the negation circuit of Fig. 10 and the regular positive MAD transfer circuit. However, there is an important difference in that the input (setting) current requirement is increased in the negation circuit. In order to switch the output core simultaneously with the main aperture during $\bigcirc\rightarrow\square$, a (relatively small) current must flow in the output loop. The input current must therefore be larger by this amount in order to switch the input. With three output circuits linking the main aperture, three units of extra current are required at the input. This can represent a significant increase in excess-current requirements, with consequent decrease in current tolerances.

Another disadvantage of all synthetic outputs is that the $A\leftrightarrow B$ operation (Fig. 8) is more critical. The circuit is repeated in Fig. 11, where three output loops (all negative) are shown connected to a single-output winding. This type of connection is permissible since the loops are otherwise totally isolated. Assume that A receives a *one* during $\bigcirc\rightarrow\square$ (*i.e.*, $x=1$) and B receives a *zero* ($y=0$). Then during the $A\leftrightarrow B$ pulse, A will cause B to set. Simultaneously, flux will switch in the output cores. In this case, the output loop currents must not only be supplied from the input to the receiver (from $A\leftrightarrow B$

transfer loop), but these currents also link the transmitter (Element A) in such a direction as to decrease the maximum allowed value of $A \leftrightarrow B$ clock pulse strength. Hence, with synthetic output loops present, not only is more excess current required in the receiver, but also less can be made available from the transmitter.

It is these disadvantages that have made a straight synthetic output approach difficult to handle. A more useful extension of the approach, however, is discussed in the following section (Section IV-G).

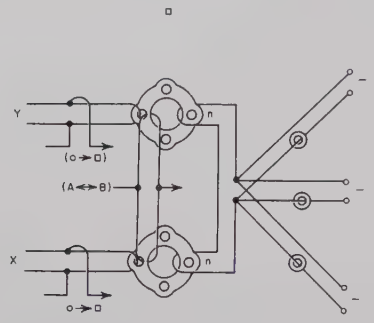


Fig. 11—Gain-doubling module with synthetic outputs.

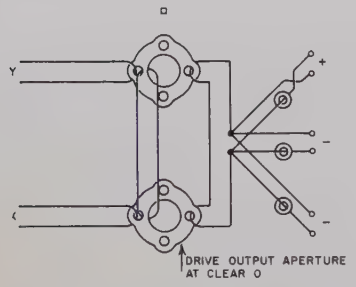


Fig. 12—Decoupled synthetic outputs.

G. Decoupled Synthetic Outputs

Much of the complexity of operation of Section IV-F may be eliminated by decoupling the synthetic output loops as indicated in Fig. 12, where two negative loops and one positive loop are shown. (It is simple to show that to achieve positive transfer with a separate output core, it is merely necessary to reverse the polarity of the coupling loop winding, as indicated.) Here the loops connect to a minor aperture of the main core, and therefore are completely decoupled during the input switching, as well as during $A \leftrightarrow B$ pulse switching. After the $A \leftrightarrow B$ operation, the output cores are switched by a separate pulse, called the set-synthetic pulse, that drives the output aperture of the main element. If the element in question is a □ element, then the Clear ○ may be used for this purpose, and conversely.

Although both positive and negative output loops can be handled in this fashion, there is an advantage to having positive output loops connect directly to their own apertures and be driven in the original manner. The main reason for this is that a synthetic positive loop conducts high currents to set the following receiver only when the transmitter is in the *one* state. With the main core in the *one* state, the high currents in the loop would tend to switch the output aperture of the transmitter instead of the receiver. To prevent the output aperture from switching, a holding mmf would be needed in the Clear direction on the inner output leg, for each positive output. This would result in considerable wiring complexity. No such extra-hold mmf is required in the negation loops, since they conduct high current to set the following receiver only when the main element is a *zero*, in which case flux cannot switch locally about the output aperture.

Another advantage of the decoupled synthetic output loops is that setting the element from an external source (*e.g.*, under keyboard control) is less critical. If an element is set by such a control current flowing through a separate input aperture, no currents will be caused to flow in any coupling loops which are so decoupled. When the clock program starts, all of the appropriate output cores are set by the first set-synthetic pulse in a normal manner. If the output loops were not decoupled, more stringent requirements would be placed on the keyboard set-up currents, to ensure adequate setting of the output cores along with the main aperture.

V. Final Circuit Form

Sections I–IV developed the philosophy of the circuit approach, but for ease of presentation, specific details were omitted. These details are now summed up in Figs. 13 and 14.

A. Positive Transfer

The circuit details of positive transfer are indicated in Fig. 13. Winding polarities and numbers of turns are indicated. Note that most windings are single-turn. Thus, the transfer loop couples transmitter elements and the receiver element with single-turn windings. The clipper core in the coupling loop is also linked by a single-turn winding. The clipper core is a stack of three standard 30-mil-by-50-mil S-5 (General Ceramics) memory cores. Stacks of memory cores are used in order to eliminate the need for manufacturing elements of nonstandard size for this application.

Assume that the module to the left is a □ module, and that to the right a ○ module. The particular □→○ clock that excites this transfer 1) links the transmitter apertures with a single-turn figure-eight winding, 2) drives the clipper core in the Set direction with a one-turn winding, and 3) biases the receiver to essentially threshold mmf through another one-turn winding. The latter statement implies that the magnitude of the clock current is nominally equal to the main-aperture threshold of the MAD elements (*i.e.*, one unit of "threshold current" flowing through a single turn provides threshold mmf.).

Subsequent to the □→○ clock pulse, states of the pair of ○ module elements are made equal by trans-

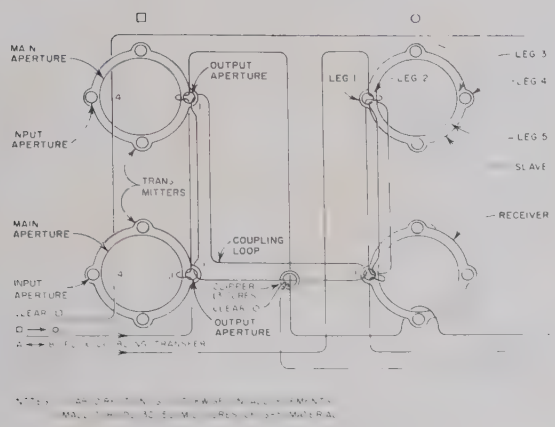


Fig. 13—Positive transfer circuit.

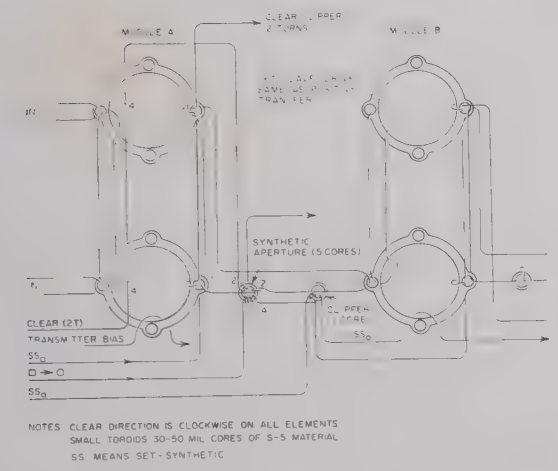


Fig. 14—Negative transfer circuit.

mitting one into the other upon excitation of the $(A \leftrightarrow B)_{\bigcirc}$ pulse. This clock current links both elements of the ○ module by a single-turn figure-eight winding about the input apertures. The transfer loop itself consists of a single-turn winding about the inner leg of each input aperture. The magnitude of this clock drive is also nominally one-threshold current.

The □ modules are then cleared by the Clear □ pulse which carries two units of threshold current through four-turn Clear windings. This generates nominally eight thresholds of Clear mmf, and results in good clearing of the elements.

Simultaneous with the Clear □ pulse is the $SS_{\bigcirc}$ (Set Synthetic) pulse (see discussion in connection with Fig. 12). This pulse clears all clipper cores in the □→○ loops (through a two-turn winding), and also excites all negative output loops of the ○ modules.

The details of this latter operation will be discussed in Section V-B on negative transfer loops.

B. Negative Transfer

The details of negative transfer are indicated in connection with Fig. 14. The negative transfer loop is similar to the positive transfer loop except for the reversed polarity of the receiver and clipper windings and the presence of a (synthetic) output core. This "core" is actually a stack of five cores, identical to those used for clippers. Note that the stack connects with the coupling loop with a two-turn winding. This is primarily to provide suitable flux capacity. If only a single-turn winding were used, the stack would have to include twice as many cores.

For negative transfer, the □→○ clock pulse drives the output core instead of the transmitter MAD's. This drive is through a four-turn winding. The drive on the clipper and receiver are identical with the positive transfer case. Again, the magnitude of clock current required is about one threshold, so that the same □→○ clock current source can be used to excite positive as well as negative transfer loops, slight wiring differences being necessary (as indicated). It can be shown that during this transfer, a Clear-direction bias on the transmitter MAD elements of about one half threshold is advantageous. Assume that the □ module holds a *zero*. Then the □→○ clock current switches the output core, which results in a loop current flow to set the receiver. This loop current tends to set the transmitter MAD. Thus, the Clear-direction bias allows the coupling loop current to be larger without unblocking the transmitters. [One-half threshold bias is more suitable than a full threshold, since in the case where the transmitter holds a *one* (*i.e.*, is Set), its major aperture threshold is considerably decreased from the case of a pure Clear state, and a full nominal threshold of bias would result in considerable flux switching about the major aperture.] If this bias were to be provided by the □→○ clock current, then the magnitude of current would have to be cut essentially in half, so that one-half threshold bias could be obtained via a single-turn winding. This would require doubling all other windings excited by this clock current. It was felt, however, that for this machine it would be advantageous to keep the □→○ current at a full-threshold level, and to provide a special, but simultaneously occurring, clock of half-threshold amplitude to provide the bias. If this special bias current is not actually applied, there is some, but not an appreciable, degradation of performance.

Subsequent to the □→○ clock current, the $(A \leftrightarrow B)_{\bigcirc}$ clock is excited. The clock arrangement and $A \leftrightarrow B$ coupling loops are exactly as described for the positive transfer.

The □ module MAD's and output cores are next cleared by the Clear □ pulse, just as in the positive transfer case.

Simultaneous with the Clear □ pulse is the $SS_{\bigcirc}$ (Set Synthetic) pulse. Thus, in all ○ modules that are arranged for some or all negative transfers on their outputs, the ○ output cores are set during this operation. This same current clears the clipper cores in the input loops of the ○ modules. The $SS_{\bigcirc}$ pulse drives the output aperture of the ○ module MAD elements with a single-turn figure-eight winding, similar to the arrangement for driving the output apertures of the transmitters during

positive transfer. Thus, any connected output aperture is driven by a single-turn figure-eight winding either to 1) effect the transfer operation during positive transfer, or 2) effect the *SS* operation for negative loops, but not both.

The circuit details for cases in which more than one transfer loop connects to a module are covered in Section V-I. Further, the coupling loops for the ○→□ transfers are identical to the □→○ loops.

C. Indicator Lights

The binary state of a module can readily be displayed by an incandescent light, without altering the information state of the module.[9] Two types of indicator light circuits are used. These are shown in Fig. 15. Consider a winding placed about an aperture. If the element is set, then large flux changes can occur with very small current through the winding. On the other hand, if the element is cleared, then essentially no flux change can occur if the winding mmf is never allowed to pass the threshold value for the large aperture. In the former case, the winding presents a high impedance, whereas in the latter it presents a very low impedance. By using the winding in parallel with an indicator lamp, the combination being fed from a sinusoidal current source (typically 250 kc), the light is energized only if the element is set, otherwise it is short-circuited [Fig. 15(a)]. Alternatively, if the winding is placed in series with the indicator, the combination being placed across a sinusoidally excited voltage source, then the lamp glows only when the element is Clear [Fig. 15(b)]. Both circuits are used in the machine. The exact circuit arrangement and indicator lamp type are shown in the figure.

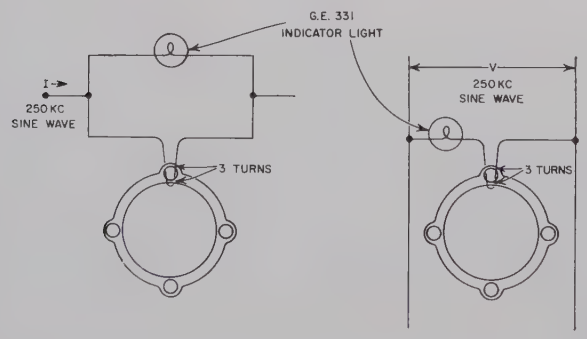


Fig. 15—Indicator light circuits.

VI. Physical Circuit Arrangement

With ten clock circuits distributed throughout the system, it is not practical to package each module individually. Individual packaging results in a large number of terminal connections and also in unnecessarily long coupling loops. Rather, it is more practical to package relatively large numbers of modules in a single plug-in package. Conventional printed wiring seems to offer no advantage for these assemblies.

It appears that the simplest and most economical packaging schemes are those based on physical alignment of a number of ○ modules and a number of □ modules (Fig. 16). With this arrangement, all clock-pulse wiring is achieved by straight passes of wire through entire rows of elements. For example, the Clear □ links the main aperture of all elements; to obtain more than a single-turn winding, several straight-through passes are made through the main apertures. The same arrangement holds for the other clock-pulse windings.

The two paired elements of a single module are placed next to each other on the board. MAD elements with four small apertures are mounted in slots in the board so that one aperture protrudes on one side of the board, and three apertures protrude on the other. Sighting down the surface of the board, we see that ○ elements on the left have three apertures above the board, while □ elements on the right have one aperture above the board (Fig. 16). This arrangement is convenient, since at most only one aperture of each element is used for input, whereas up to three apertures of each element may be used for output. Below the board are the $(\square\rightarrow\bigcirc)^{1,2,3}$ clock-pulse windings, as well as Clear □ and $(A\rightarrow B)_{\bigcirc}$. The other five clock windings are located on the other side of the board.

All of the circuit boards for the machine are mounted in a single plane to facilitate demonstration and debugging. This arrangement is apparent in the photograph of Fig. 17. Even though the total area of the circuit boards is large, the volume is small since the maximum thickness dimension is little more than half an inch. For ease of construction, no special effort was made to conserve area.

With the MAD elements lined up in three dimensions, it is difficult to present the wiring details in simple fashion. It is of considerable help to illustrate the MAD elements in distorted fashion, as indicated in Fig. 18. Thus, all apertures of each element are in separate vertical columns, so that the clock windings personal to each aperture can be indicated without confusion. Furthermore, the three apertures available for use as outputs are shown together on one side of the element and separate from the single-input aperture.

It has already been indicated that each transfer loop contains one clipper core and one (synthetic) output core where negation is necessary. Simplicity of assembly is achieved by placing these cores at the receiver end of the coupling loops, opposite their receiver apertures, rather than at the transmitter end where there may be as many as three separate coupling loops converging. Each input can be excited by any one of three Advance pulses. Since there may be one clipper and one negation core per loop, there are, at most, six categories of small toroids. These toroids are lined up with not more than

[9] J. A. Rajchman and A. E. Lo, "The transfluxor," Proc. IRE, vol. 44, pp. 321–332; March, 1956.

six strings, and are placed adjacent to their corresponding input aperture (Fig. 18). All clock-pulse lines that connect to these toroids are strung straight through the chain. The only individual windings required are in the coupling loops, four of which are indicated in Fig. 18.

Assume that the left row of modules in Fig. 18 is □ modules. Two such modules are indicated. Assume that the upper module, labeled $\square_x$, stores the state of variable x, and that the lower module, labeled $\square_y$, stores the state of variable y. To the right are the ○ modules. The circuit is arranged so that upon completion of the clocks $(\square\rightarrow\bigcirc)^{1,2,3}$, the upper ○ module holds the state of logical function $x+y$, and the lower module holds the state of the function $x+\bar{y}$.

In the particular case illustrated, the variable x is transmitted to three receivers during the $\square\rightarrow\bigcirc$ clocks (*i.e.*, three positive transfers are involved). Each positive transfer loop connects to a separate aperture of the $\square_x$ module. On the other hand, module $\square_y$ is arranged to transmit two negative outputs at $(\square\rightarrow\bigcirc)^1$ and $(\square\rightarrow\bigcirc)^3$, and one positive output at $(\square\rightarrow\bigcirc)^2$. Both negative outputs connect via a single output aperture, as in Fig. 12. Thus, only two output apertures are used in the $\square_y$ module, one for the positive and one for the two negative transfers, whereas all three of the output apertures of the $\square_x$ module are required, one for each positive transfer.

Consider a positive transfer loop, for example, the upper transfer loop shown in the figure. (The details of physical construction of coupling loops are discussed in connection with Fig. 19.) This particular loop is excited by the $(\square\rightarrow\bigcirc)^1$ clock. The specific manner in which the clock current is made to excite various portions of the transmitter and receiver elements is indicated schematically beneath the circuit arrangement. Consider the horizontal line labeled $(\square\rightarrow\bigcirc)^1$. An arrow at any position indicates that this current is made to link the corresponding leg of the devices lying vertically above a

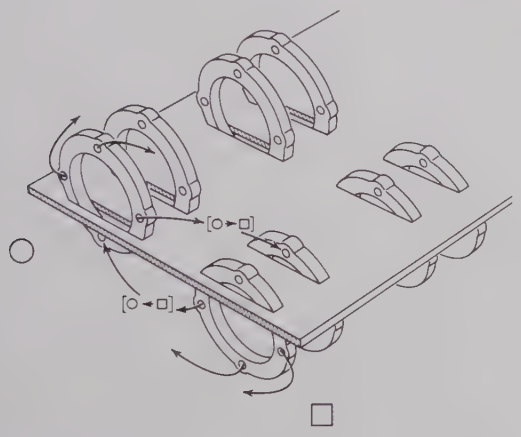


Fig. 16—Three-dimensional element layout-perspective.

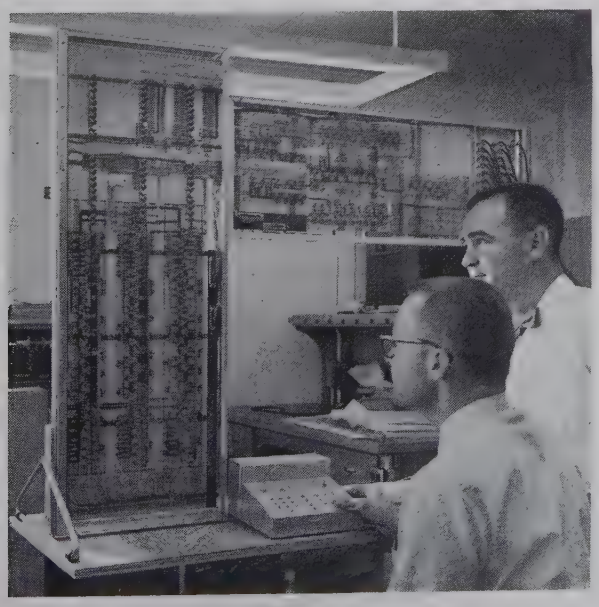


Fig. 17—Photograph of the all-magnetic computing system.

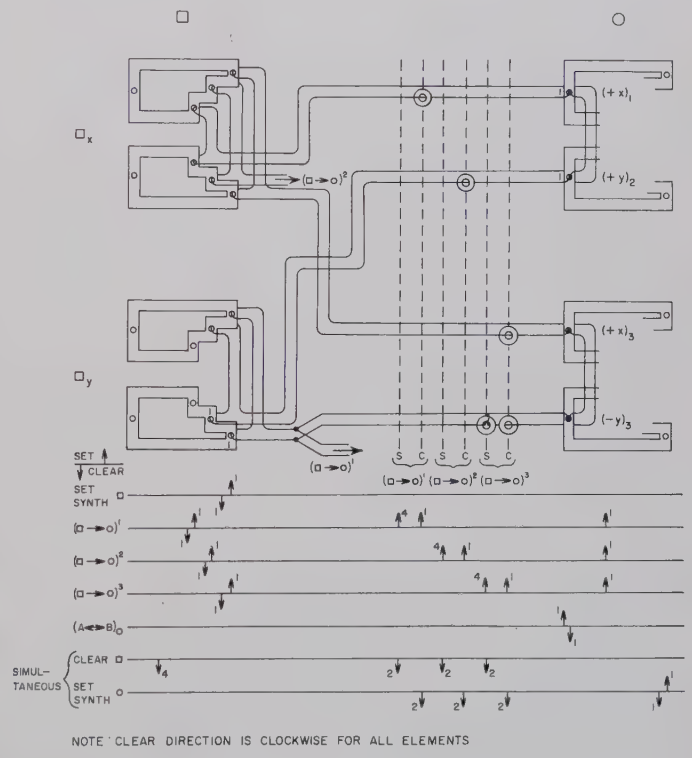


Fig. 18—Positive and negative transfer circuit detail.

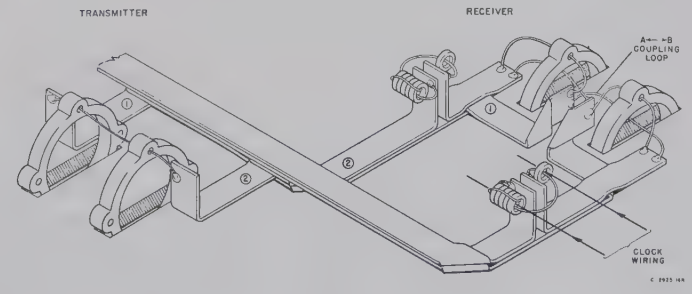


Fig. 19—Coupling loop construction.

line defined by the arrowed segment. An arrow pointing upward indicates that the current is made to link that leg in such a manner as to tend to cause that leg to switch in its Set direction. A downward arrow implies that the current tends to drive that leg in its Clear direction. The number adjacent to the arrowed line indicates the number of turns on the linking winding. Thus, following the $(\square\rightarrow\bigcirc)^1$ line from left to right, we see that it first connects with the legs adjacent to the output aperture farthest left, driving the outer leg upwards and the inner leg downwards. This type of driving arrangement is equivalent to the *one-one* figure-eight transmitter drive of Fig. 13. Continuing along the $(\square\rightarrow\bigcirc)^1$ line, we see that it drives the clipper core in its Set direction (through a single turn), and biases the receiver essentially to threshold (again through a single turn). This circuit arrangement is clearly identical to that developed for the positive transfer in Fig. 13.

Also at $(\square\rightarrow\bigcirc)^1$ time, the $\square_y$ module is indicated as delivering a negative output to some $\bigcirc$ module not indicated in the figure.

At $(\square\rightarrow\bigcirc)^2$, $+x$ is delivered to some $\bigcirc$ module not indicated, and $+y$ is delivered to the upper $\bigcirc$ module. These positive transfer loops are identical with the positive transfer loop already described, except that they are excited by the $(\square\rightarrow\bigcirc)^2$ pulse.

Finally, at $(\square\rightarrow\bigcirc)^3$ both x and $\bar{y}$ are delivered to the lower $\bigcirc$ module. The positive transfer loop is as previously described. Further, it is clear that the arrangement of the negative transfer loop agrees with the arrangement developed in Fig. 14.

Note that it is not necessary for inputs to the same module to occur during the same clock pulse. Thus, the inputs to the upper $\bigcirc$ module occur at different times, whereas the inputs to the lower $\bigcirc$ module are simultaneous.

Also, it must be noted that not all $\square\rightarrow\bigcirc$ windings are achieved by straight-through passes of wire. This is mainly due to a shortage of available apertures. Thus, with MAD's having five small apertures, rather than four, the three positive output positions could have been vertically isolated from the single negation position. As it is, the output apertures to the far right in the figure are used for positive or negative output. Therefore, the $SS_{\bigcirc}$ pulse links only the $\square_y$ module with a figure-eight winding, and the $(\square\rightarrow\bigcirc)^3$ pulse links only the $\square_x$ module with a figure-eight winding. In general, the *SS* and $(\square\rightarrow\bigcirc)^3$ clock wires, as they are threaded through the line of modules, must be made to link only the appropriate modules.

Others of the clock windings also deviate from purely straight-through passes. This results from our initial philosophy of conservative design. Thus, consider the $(\square\rightarrow\bigcirc)^{1,2}$ lines. As indicated, they drive only apertures which are either unused, or connect with positive transfer loops. Although, in principle, there is no effect from having an unused aperture driven by the Advance pulse, there is slightly improved performance by eliminating this unnecessary operation. Hence, in stringing these lines, they are purposely arranged to miss unused apertures along the line.

Similarly, consider the case of receiver bias. Any single MAD receives data, at most, at only one of the three Advance times. In principle, there is no effect from having the receiver biased at all three Advance times; however, there is again some small degradation, and the wiring is arranged so that each receiver element is biased only at the time that it is actually used.

On the other hand, all clock wiring of the separate toroids is achieved by pure straight-through passes.

Although not explicitly indicated in the figure, the considerations for transfer in the $\bigcirc\rightarrow\square$ direction are precisely identical to those described in connection with the $\square\rightarrow\bigcirc$ transfer.

A. Coupling Loop Construction

The only mechanical assembly details worthy of special note are in connection with the coupling loop assemblies. In Section IV-B, it was indicated that a specific clipper core is inserted in each loop in order to suitably tailor the flux-gain characteristics. Further, to achieve best *one* transfer, it was attempted to attain the lowest reasonable loop resistance and inductance. This again fits the philosophy of conservative design for this feasibility machine.

To achieve low-loop impedance, the portion of the loop between transmitter and receiver modules was made of copper strip-line. These strips were made by sandwiching a strip of Scotch tape between 3/16-inch strips of 5-mil-copper shim stock. The details of the arrangement are indicated in Fig. 19. For the case shown, there is one (transmitter) module to the left, and one (receiver) to the right. Both inputs to the receiver module are negative. This is clear from the sketch since each loop contains a (synthetic) output core as well as a clipper. Further, these inputs occur at different clock times, as is evident from the difference in lateral positioning of the cores (see Fig. 18).

Each strip line was brought close to its termination point at the modules. At that point, regular (circular) wire is used to couple the MAD. Also, regular wire is used to couple the small toroids to the strip line (as shown). Clearly, this leads to more solder connections per loop than there would be if a single regular piece of wire were used to string all loop elements together. Further comment on this point is made in Section VIII.

The method of inserting the input loops and the $(A\leftrightarrow B)$ coupling loop is somewhat different than that indicated schematically in Figs. 13 and 14. In that case, the loops independently thread the input apertures, requiring therefore two separate wires through each input aperture. To conserve room in the aperture (the elements used had relatively small apertures—nominally

18 mils diameter), the scheme indicated in Fig. 19 was adopted. In this case, only a single wire passes through the aperture, and the method of connection is such that the input loop couples to this wire in such a manner as to link the outer leg about the input aperture, and the $A \leftrightarrow B$ loop couples to the wire in such a manner as to link the inner leg. The sharing of a common wire in this way, with the resultant direct connection of the loops, is possible only because 1) the loops are not directly connected at any other point in the circuit, which of course would result in a short circuit, and 2) the input and $A \leftrightarrow B$ loops terminate at the input aperture with precisely the same number of turns—namely one. This scheme again results in more solder connections per loop than would be required by the more direct scheme of isolated loops.

VII. Clock Generator Requirements

As stated in the text, the current pulse amplitudes required are nominally one and two (main aperture) thresholds. (A threshold current through a single-turn winding generates a threshold mmf.) The amplitude requirement, therefore, is determined by the element size and material, as well as by the circuit configuration.

A nominal pulse width of 5 μsec was initially specified. Pulse widths as narrow as 1 to 2 μsec could probably have been used, but in view of possible system problems (drive line transients and rise and fall-time deterioration along the line), it was decided, in the initial spirit of conservative design, to design for a wider pulse, thereby minimizing any possible effects.

An empirical circuit design method was developed which predicted a pulse amplitude range of somewhat less than ± 25 per cent for clock pulses driving circuits made up of small numbers of modules, such as one- or two-bit shift registers. Ranges of ± 20 per cent were achieved experimentally. The pulse amplitude range for the complete machine is ± 10 per cent. These ranges are essentially for the "worst case," since ranges were checked with all clock-pulse amplitudes varied simultaneously from one end of the range to the other.

The voltage requirement on each clock-pulse line is approximately 200 volts. This voltage drop was primarily across the resistance in the small-drive line wires (#40 wire), the voltage due to the switching of the elements being small in comparison. The situation would be quite different if elements with larger small-apertures had been used, assuming that wire diameter is correspondingly scaled, since the resistance of the wire, and therefore the required voltage, vary inversely as the square of the small-aperture diameter.

The maximum pulse repetition rate of the system was determined by the particular laboratory pulser that was used. Ideally, with a clock generator designed specifically for this machine, a 20-kc rate could have been obtained, that is, ten 5-μsec sub-clocks in 50 μsec. The particular pulser used with this machine limited the clock rate to 3 kc. Additional requirements are placed on the clock generators by drive-line ringing. To reduce ringing, the pulse rise time was increased to about 1 μsec and the fall time to 3 μsec.

While no work was done in designing special clock drivers for this system (laboratory pulser was used for clock source), it is apparent that the voltage and current requirements are well within the capability of commercially available *p-n-p-n* semiconductor switches. It appears that a clock generator system based on such switches would be relatively small and inexpensive.

The fact that certain system effects (*i.e.*, drive-line ringing, element variation, etc.) would decrease the level of performance of the machine below that for the individual modules was predicted, but no attempt was made to predict the magnitude of the various effects. The drive-pulse amplitude range is considered to be a good indicator of the general over-all level of performance of these circuits. As already indicated, the pulse amplitude range for the complete machine was ± 10 per cent, although ± 20 per cent range was achieved for the individual module. The most significant system effect is the ringing on the drive lines. Lengthening the rise and fall times reduced these transients to the point where the machine operated with the indicated ± 10 per cent range. It may be noted, however, that no special attention was paid to wiring arrangements for best transient response, although it is clear now that significant improvement can be obtained. No effort was made to measure the effects on range of other system effects since it was apparent that they were of minor importance compared to the drive-line transients problem.

It is thus apparent that this circuit technique could be applied to much larger systems as long as coupling-loop impedance is kept sufficiently low and drive-line transients are kept within bounds.

VIII. Conclusions

The logical module developed in detail herein is used as the sole logical building block in the decimal arithmetic computing system to be described in Part II. As indicated in Section VII, the performance of the completed system was essentially as predicted; that is, no unforeseen difficulties were encountered, and within the framework of logical interconnection rules, modules could be interconnected in any desired fashion to synthesize the necessary logical functions. This is an important result, since the detailed analysis of these all-magnetic circuits is quite difficult, and one could not be absolutely sure that some undesirable, unforeseen coupling or interaction would not occur among modules of a relatively large net. In this sense, at least for this system, it can be unequivocally stated that the empirical design procedures "in the small" (*i.e.*, at the module level) hold for systems "in the large."

The reliability of systems of this type should be excellent. For all practical purposes, the basic elements (ferrite and wire) have indefinitely long life. Reliability is therefore limited by 1) wire insulation, 2) solder connections, and 3) clock drivers. As far as solder connections are concerned, with the three-dimensional assembly techniques adopted here, where clock windings are achieved by (straight) push-through techniques, there are basically only two solder connections per clock, per board. Therefore, the more modules per board, the less the total number of solder connections per system. The other "source" of solder connections is in the coupling loops. In the present scheme, where negation is obtained by separate output cores, and where each coupling loop has a separate clipper core, the number of solder connections per loop is relatively large. However, by appropriate element design, both of these functions could be achieved in a single multiaperture element, thereby eliminating the need for a large number of solder connections. In general, of course, the more logical functions per device, the less the number of coupling loops, and hence, the fewer solder connections required.

As for drivers, it was indicated in Section VII that this entire system could have been driven by a relatively simple array of commercially available *p-n-p-n*-type semiconductor switches. For even slower operation, it may be quite possible to adapt some type of mechanical commutator for clock power. In the latter case, not only may the final system be extremely conservative with regard to power consumption, but it might be very resistant to radiation damage as well.

As far as the cost of such systems is concerned, it is clear that the components (ferrite and wire) are inexpensive. The major costs then are in the assembly and drivers, yet it seems possible to construct drivers of considerable simplicity and, hence, low cost. The question of assembly requires more study. While there was considerable initial concern over the possibility of being able to even construct such a system, the assembly was relatively simple and straightforward, although tedious. However, it must be recognized that the finished system cannot be considered as a highly complex or highly interconnected one, so that extrapolation may be misleading.

In general, it seems that all-magnetic logic systems of the type considered here may be well adapted to digital computing system realization where speeds are limited to, perhaps, 100 kc. One of the interesting features of such a system, aside from the high potential reliability, is that it appears to offer an answer to the problem of matching the system to an environment in which large amounts of communication between system and environment are required (generally referred to as the "interface" problem). In this regard, the ability to interrogate the magnetic elements without disturbing their state may be very important. In particular, the readout process can involve the control of a significant amount of power. In the present case, for example, with no intervening elements, commercially available 70-mw incandescent bulbs were nondestructively controlled by the MAD elements. In fact, two or three times this much power could have been controlled in this fashion.

However, the field of all-magnetic logic is very young, and although one particular approach has been detailed here, there are many other approaches possible, some of which hold promise for even more interesting system structures. A number of such possibilities are now being studied at Stanford Research Institute, as well as at other places.

Acknowledgment

The final circuit approach developed here has been an evolutionary process and incorporates a number of ideas from other people, in particular Drs. D. R. Bennion and D. C. Engelbart. The authors are also indebted to C. H. Heckler, who developed techniques for testing the magnetic elements; J. A. Baer for the input-output work; and to W. K. English, without whose devotion the machine might never have been completed. Acknowledgment is also made of the excellent work done by J. Hunt in wiring the machine. Thanks are due also the Cambridge Air Force Research Laboratories for permission to publish this paper.

Design of an All-Magnetic Computing System: Part II—Logical Design*

H. D. CRANE†, SENIOR MEMBER, IRE

Summary—A logical design technique is developed for use with the particular module developed for this system. The detailed properties of this module, as well as the philosophy that led to its particular form, were covered in Part I of the paper. Briefly, the module forms the (inclusive) OR function of two input variables. This function can subsequently be transmitted to three receivers, each transfer being independently logically positive or negative. The read-outs are nondestructive and the transmitter module must be explicitly cleared before read-in is again possible. In view of the relatively small fan-in and fan-out for this module, and since only the OR function can be directly formed during any single transfer, complex logic functions must be formed slowly, a step at a time. This step-by-step generation of functions results in the need for more modules than might otherwise be required, but aside from that, the synthesis techniques are not particularly different from those of customary logical design. In particular, the design of an arithmetic unit designed for decimal addition, subtraction and multiplication is outlined. Some comparisons are noted between this particular all-magnetic logic scheme and conventional core-diode schemes. Comparisons are also made between magnetic logic schemes in general and some other realization schemes, such as ac-operated parametrons and conventional transistor systems.

Introduction

IT WAS desired to construct a small digital logic system using all-magnetic logic circuits, in an attempt to demonstrate the feasibility of constructing systems larger than small shift-register arrays, counters, and so forth, to which roles all-magnetic logic circuits have been limited to date.

In somewhat typical fashion, this "feasibility" system took the form of a decimal arithmetic unit; one with the ability to add, subtract, and multiply decimal numbers.[1,2] The design called for the use of about 325 logic modules of two MAD's (multiaperture devices) each. The system is controlled from a manual keyboard, and read-out from the machine is via incandescent lamps operated directly from the MAD elements, no intermediary elements being required.

In view of the nature of this feasibility study, no attempt was made to optimize the system design in the sense of reducing the element count or maximizing speed. Instead, a rather straightforward design was adopted, in order to reserve the majority of effort for the circuit design and assembly techniques. Although the design used here is based upon the usual "module" approach, the properties of the single-module type employed are somewhat different from those normally encountered in the design, for example, of transistor computers; and a brief review of the formal internal structure of any computing system may be of some help in establishing a common take-off point.

A simple representation of any sequential logic system is indicated in Fig. 1.[3] It recognizes that there are two distinct types of "components" required, namely, storage components and logic components. In the figure, all the storage components are schematically grouped together under the label "memory;" all the logic components are grouped together under the label C.L. (combinational logic). At any instant the *state* of the system is determined by the state of the storage elements (there is assumed to be no significant storage associated with the logic elements). Of course, in any practical system there are "external" inputs and outputs to contend with also, but those are not treated here, since the main purpose is to review the basic structure of the internal organization of a computer.

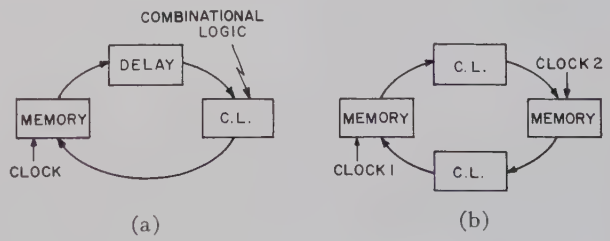


Fig. 1—General representation of a synchronous logic system. (a) Single-clock system. (b) Two-clock system.

Let us treat synchronous binary logic systems only. Consider the system of Fig. 1(a). At each clock time the state of the storage elements is altered to a new state. This new state (or "next" state, as it is generally called) of any particular storage element is determined by a logical combination of the present state of a selected set of the storage elements. Thus, the relation between the

* Received by the PGEC, October 31, 1960; revised manuscript received, January 18, 1961. The work reported here was carried on at Stanford Research Institute, Menlo Park, Calif., under the sponsorship of the AF Cambridge Research Labs., Bedford, Mass., Contract AF 19(604)-5909.

† Stanford Research Institute, Menlo Park, Calif.

[1] H. D. Crane, "Research on the Realizability of an All-Magnetic Computing System," Stanford Res. Inst., Menlo Park, Calif., Rept. No. 1, Contract AF 19(604)-5909; October, 1959.

[2] E. K. Van De Riet and C. H. Heckler, Jr., "Research on General Digital Logic Systems Utilizing Magnetic Elements and Wire Only," Stanford Res. Inst., Menlo Park, Calif., Final Rept., Contract AF 19(604)-5909; October, 1960.

[3] D. A. Huffman, "A Study of the Memory Requirements of Sequential Switching Circuits," Res. Lab. of Electronics, Mass. Inst. Tech., Cambridge, Tech. Rept. 293; March 15, 1955.

set of next states and the set of present states is determined by the specific arrangement of the combinational-logic network.

An important observation here is that in the one-clock system of Fig. 1(a), a minimum delay must be present in the "loop." This is necessary to ensure that during any clock pulse, none of the signals on the lines controlling the setting and resetting of the (binary) storage elements are altered. If this condition were not met, then some storage elements might tend to change state again during the *same* clock pulse. (Although it may be possible to design systems in this manner, they would be very sensitive to variations in specific time delays, and so forth.) Single-clock systems of this type have been used in core-diode systems, where capacitors are usually used for delay, as well as in transistor (and vacuum tube) logic systems.

An alternative to the single-clock system, where delay is required in each control loop, is a two-clock system in which *two* separate groups of storage and logic elements are used, as suggested in Fig. 1(b). The two clocks are arranged so that their pulses are interspersed in time. Thus, at clock C_1, the left storage elements are altered to their next state, under control of the state of the right storage elements. There is no delay required in this operation, since the right storage elements themselves cannot be affected except during clock C_2, which follows C_1 after some minimum time to allow the left storage elements to reach steady state. During clock C_2, the right storage elements are altered, under control of the states of the left storage elements, and so on. Thus, two-clock systems do not require explicit delay, and any parasitic delays which tend to slow up the switching of the storage elements are accounted for by merely increasing the period between clock pulses. All dc-coupled logic systems which do not use reactive elements for delay or storage are necessarily two-clock systems. (If for some reason it is advantageous to have more than two groups of storage elements, then clock systems involving more than two clocks can be used in an extension of the spirit of the two-clock system—delay again not being required.)

All-magnetic logic circuits cannot operate on a single-clock system, since there are no elements available for delay between the magnetic storage elements. On the other hand, although the circuits evolved in Part I of this paper involve the use of considerably more than two clocks, this does not alter the fact that *logically* the evolved system is of the two-clock type; let us consider the nature of some of these "extra" clocks. There are two groups of storage elements, labelled □ and ○. During the □→○ pulse, the states of the ○ elements are altered under control of the □ elements; or, stated somewhat differently, the □ modules transmit logical combinations of their own states to the ○ modules. However, the transfer from the □ to the ○ elements is nondestructive to the states of the □ elements. Therefore, before the next ○→□ pulse, the □ elements are explicitly cleared by a Clear □ pulse. By symmetry there is also, therefore, a Clear ○ pulse, and the clock cycle takes the form

$$\cdots, \square \rightarrow \bigcirc, \text{ Clear } \square, \bigcirc \rightarrow \square, \text{ Clear } \bigcirc, \cdots.$$

Further, however, it was indicated in Part I that because of the nondestructive nature of the read-out from an element, it is possible to increase the "fan-out" capability of each element by transmitting from that element at three different times, each time transferring to a different receiver. Thus, each "advance clock" is replaced with three such clocks, and the clock system is evolved to the form

$$\cdots, (\square \rightarrow \bigcirc)^1, \quad (\square \rightarrow \bigcirc)^2, \quad (\square \rightarrow \bigcirc)^3, \quad \text{Clear } \square,$$
$$(\bigcirc \rightarrow \square)^1, \quad (\bigcirc \rightarrow \square)^2, \quad (\bigcirc \rightarrow \square)^3, \quad \text{Clear } \bigcirc, \cdots.$$

Still, the logic is basically of the two-clock type.

Logic Properties of Module

Part I of this paper is concerned primarily with the details of developing a suitable module for use in constructing the machine. The final module form is indicated schematically in Fig. 2(a). Briefly, each module contains two MAD elements. One input variable connects to each element. After the input cycle, the two elements of each module "transmit to each other," in order to form the inclusive OR function, both elements of a module ending up in the identical state. Let us define a Set MAD as being in the *one* state, and a Clear MAD as being in the *zero* state. Assume that the module is in the *zero* state (*i.e.*, both elements of the module are clear). After input excitation the module is set to the *one* state if $x=1$, or $y=1$, or both—that is, the module is arranged to form the (inclusive) OR function $(x+y)$. Subsequently, the stored function $(x+y)$ can be read out to a maximum of three receivers. But each transfer can be made logically positive or negative, so that the function $(x+y)$, or $\overline{(x+y)}=\bar{x}\bar{y}$ can be independently transmitted to each receiver. Note that the latter function is just the regular NOR function (*i.e.*, not OR).

An important property to note here is that the inputs can only cause switching to the *one* state, but not in the opposite direction. Thus, an explicit Clear operation is provided for unconditionally returning the module to the *zero* state. To arrange such modules in a logical two-clock system, the clocking arrangement as previously developed, is

$$\cdots, \square \rightarrow \bigcirc, \quad \text{Clear } \square, \quad \bigcirc \rightarrow \square, \quad \text{Clear } \bigcirc, \cdots$$

where □ and ○ are the names assigned to the two groups of modules and □→○ and ○→□ are the Advance pulses. (In the actual system, each Advance "clock" is really a set of three clocks in order to achieve the appropriate fan-out, but that is not important here.)

For the reader who is better acquainted with the more standard logical realization schemes using transistors, tubes, diodes, etc., it is of interest to indicate the form

that an equivalent module might take. Thus, in Fig. 2(b) it is indicated that the module contains a conventional flip-flop, with logical Set circuitry only. Assume that the flip-flop is reset. During the clock pulse, the flip-flop is Set, or not, depending upon the state of the input control function $(x+y)$. Hence the flip-flop may be said to store the state of the function $(x+y)$. Both "sides" of the flip-flop are available for output, so that $(x+y)$, or $(\overline{x+y})$ may be read out. To be consistent with the properties of the magnetic module, the total number of loads must be limited to three. Finally, a reset clock pulse unconditionally resets the flip-flop. The clock sequence for a system made from these modules would be identical with that for the magnetic modules, where "Reset" plays the role of Clear, and "Clock"plays the role of Advance. (Note that normally, where both Set *and* Reset logic circuitry is provided for each flip-flop, a system actually involving only two clocks is possible.)

The system to be described is made *exclusively* of these modules. In the figures that follow, a module is indicated simply as a ○ or a □. A coupling loop is indicated by an arrowed line starting at the transmitter and terminating at the receiver. If the loop is arranged for negative transfer, then a dot is placed on the line near the transmitter. As a mnemonic device, the dot may be thought of as the extra (synthetic) output core required for negative transfer. Further, because it is not permitted to read into and read out of a module simultaneously, it is a basic rule that □ modules can transmit to ○ modules only, and vice versa.

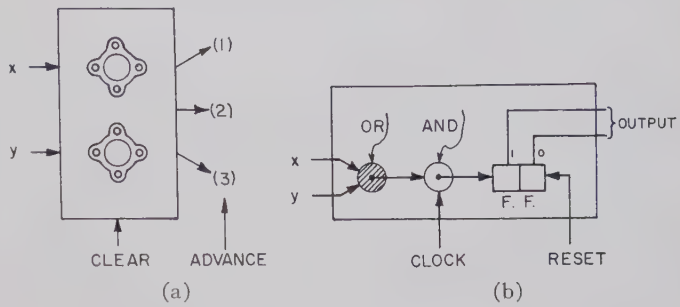


Fig. 2—Basic module. (a) Two-input MAD module forms $x+y$. The function $(x+y)$ or $\overline{x+y}=\bar{x}\bar{y}$ can be transmitted independently to three receivers during three separate advance clocks. (b) Logically equivalent transistor-style module.

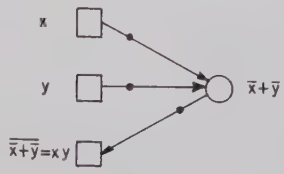


Fig. 3—Forming the logical product (AND) of two variables.

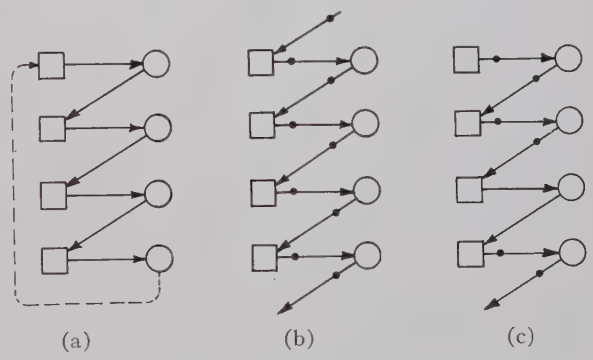


Fig. 4—Basic shift-register arrangement. (a) All positive transfers. (b) All negative transfers. (c) Mixed transfers—positive or negative by pairs.

Special Logic Configurations

Indicated here are some special module configurations of which considerable use is made.

Gating (Logical AND)

Gating usually implies an AND gate (*e.g.*, a two-input AND gate), the output of which is *zero* if one of the inputs (the control input) is *zero*, and the output is identical to the other input if the control input is *one*. Although the basic module used here generates the OR function, the AND function is easily achieved by invoking the well-known logic relation $\bar{x}+\bar{y}=xy$. Thus, given x and y in □ modules, xy can be formed as indicated in Fig. 3.

Shift Register

An ordinary binary shift register is indicated in Fig. 4(a). The modules are connected in a chain, one connected with the next by a positive transfer. It is important to keep in mind that the ○ modules and □ modules are physically identical; they are distinguished only by their placement relative to the clocking system. Thus, Clear □ connects only to □ modules, and so forth. The register may be arbitrarily long. Furthermore, it can be closed on itself (as indicated by the dashed line in the figure) to form a circulating register in which a pattern once entered circulates indefinitely.

Although all the transfers in the register of Fig. 4(a) are indicated as positive, they could just as well be negative. Thus, consider a bit in any □ module. If the subsequent □→○ transfer and ○→□ transfer are both negative, then the particular bit ends up in the adjacent □ module in precisely the same form as though it experienced two positive transfers. (This is true, of course, since $(\bar{\bar{x}})=x$.) In fact, without effect on the information carried by the □ modules, adjacent pairs of transfer loops could be independently positive or negative, as suggested by the arrangement of Fig. 4(c).

(Logical) Flip-Flop

Although each module provides storage, a module cannot be used for long-term storage. Long-term storage

is used here in the conventional sense of placing a flip-flop in a certain state and having it remain in that state for an indefinite period, until specifically altered. Although each of the MAD elements of a module exhibits storage, it must be treated as "temporary storage" since each element is unconditionally reset (or cleared) during *every* clock cycle. In this case, to achieve long-term storage, a pair of modules is required, the pair being arranged in the form of a one-bit shift register. Consider the one-bit shift register of Fig. 5(a). Although such a network can stably store a *one* (set state) or *zero* (clear state) indefinitely, it is impossible to clear such a network compatibly from other similar logic elements. This is so because each module forms the logical OR function, so that if any input is a *one* the output cannot be made *zero* regardless of the state of the other input. Thus, external inputs to either element can only cause the circuit to assume the *one* state, but NOT the *zero* state. However, if the modules are connected by negative transfers [Fig. 5(b)], then Set (S) and Reset (R) connections can be properly made. In this network, the two stable states refer to the condition where one loop transfers a *one* (high level of flux) and the other a *zero* (low level of flux). It is clear that these patterns represent stable states since starting at any point and tracing around the closed circuit, an even number (namely two) of negative transfers are encountered before returning to the starting point. This particular circuit is defined here as an *FF* (flip-flop). Whenever the S line is a *one*, the *FF* switches to the Set state, if it is not already in that state. Similarly, whenever the R line is a *one*, the *FF* assumes the Reset state.

In the schematic drawings to follow, a rectangular symbol enclosing a single ○ and a single □ module represents an internal interconnection of the negative-negative type, although for simplicity the internal connections are not actually shown [Fig. 5(c)].

Binary Counter

Conventionally, a counter or scalar is a flip-flop with a single input arranged so that the flip-flop changes state at each input pulse. Where that function is required in the machine, the circuit of Fig. 6(b) is used. The circuit is shown in logical form in Fig. 6(a), where the symbol ⊕ is used to indicate Exclusive OR. Given inputs x, c, then output is $z=x\oplus c=x\bar{c}+\bar{x}c$. Consider c to be a control input. As long as $c=0$, then $z=x$, and the "present" state of x is circulated (stored) indefinitely. Whenever $c=1$, then $z=\bar{x}$, and the state of the storage is reversed. Since the Exclusive OR function cannot be provided by a single module, it must be synthesized as in the network of Fig. 6(b). Assume that the variables x, c are available in □ elements. Then, during □→○ the function $(\bar{x}+c)$ is advanced to the top ○ element, and the function $(x+\bar{c})$ to the lower. (Recall that a dot on the transfer line indicates negative transfer.) During the subsequent ○→□ pulse, the negative of these functions is advanced to the top □ element. Since $\overline{x+\bar{c}}=\bar{x}c$, and $\overline{\bar{x}+c}=x\bar{c}$, then the function formed is $\bar{x}c+x\bar{c}$, which is exactly the Exclusive OR function sought. Hence, each time $c=1$, the state of the storage is reversed.

Fan-Out

Each module has a fan-out capacity of three—*i.e.*, each module can control three receivers. Effectively increased fan-out can be attained for controlling a set of gates by cascading modules as in Fig. 7. During one complete clock cycle, one variable may fan-out to nine receivers. Thus, during the first half of the clock cycle (*e.g.*, □→○), three ○ modules (receivers) may be controlled. During the second half of the clock cycle ○→□, these three ○ modules may, in turn, control nine □ modules, and so on.

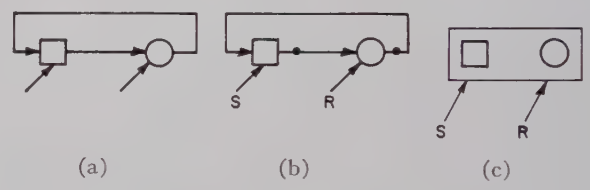


Fig. 5—Logical flip-flop circuit. (a) One-bit shift register with positive transfers cannot be set and reset. (b) Settable and resettable flip-flop requires negative transfers. (c) Symbol for circuit (b).

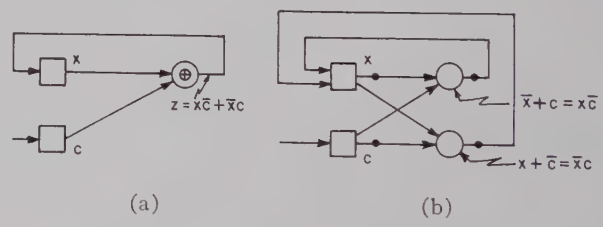


Fig. 6—Binary counter circuit. (a) Schematic. (b) Realization with present modules.

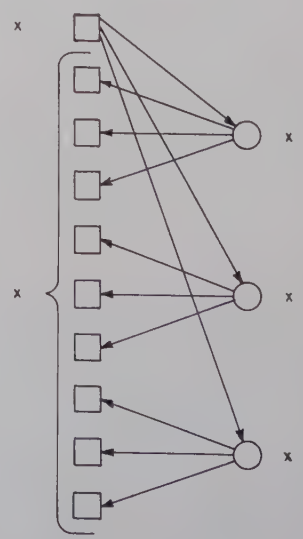


Fig. 7—Fan-out arrangement: 1→3→9.

Comparison of Magnetic Logic Systems to Some Other Systems of Logic

Magnetic logic systems, whether core-diode[4] or all-magnetic, operate from clock pulses. For proper perspective, it may be appropriate to compare magnetic logic systems with some other synchronous systems that operate on dc power (such as conventional transistor systems) and those that operate on ac power (such as parametrons).[5] The intention here is not to develop any formal relationships between the various schemes but rather to make "passing comments."

In dc systems, the clocks do not play any significant role with respect to supplying power; the power comes from the dc sources. In magnetic systems, on the other hand, all of the power comes directly from the clocks (aside from dc-bias current supplies in some systems).

Another point of comparison is the technique of binary variable storage. In this regard, dc systems and magnetic systems are again at opposite ends of the spectrum. In the former, variable storage (in conventional flip-flops) is associated with the continual dissipation of energy. In magnetic systems, "memory is static" and there is no energy dissipated between clocks. Further, there may be indefinitely long periods between clocks with no effect on the logical manipulation. (It is this property, of course, that makes magnetic systems so appealing for lower-speed systems where power is at a premium.)

Systems like the parametron lie midway in the spectrum from dc systems to magnetic systems. Thus, all power comes from the (sinewave) clocks. On the other hand, variable storage is associated with continuous energy dissipation.

[It may be of interest to note that a relay logic system, in which mechanical latching is used (*i.e.*, binary variables stored in latching relays), is similar to a magnetic system with regard to variable storage and energy properties.]

Another point on which "conventional" magnetic systems differ is with respect to gain properties. ("Conventional" is used here to imply core-diode schemes, although the arguments on gain also apply to the all-magnetic logic system scheme used in the present system.) Consider again the schematic arrangement of Fig. 1. In conventional dc systems, as well as in ac-operated parametrons, the memory "elements" are inherently bistable. This situation is depicted in Fig. 8(a) for the case of a two-clock system. Thus, during any single clock time the only requirement on the connecting logic circuitry is that the control signal at the memory elements be greater or less than some critical value. After initial triggering, the devices go the rest of the way "on their own," towards either stable state. Magnetic elements, on the other hand, have essentially an infinite number of stable states. In that case, bistability is obtained by proper adjustment of the flux-gain characteristics around the entire loop, including memory and logic "elements" [Fig. 8(b)]. Here too, of course, there is a critical point above which the flux tends towards one state and below which it tends towards the other state. (See Figs. 4 and 5 of Part I.) For example, consider the magnetic flip-flop of Fig. 5 (this paper). Neglect external inputs for the discussion here. If the flux state of the □ module is arbitrarily set to a level just above the critical level, then as a result of the □→○ clock, the subsequent flux state of the ○ module is higher than that of the initial state of the □ module. After ○→□, the flux state of the □ module is still higher, and so on. As a result of each transfer, the flux state increases towards its stable state. (Compare this with the case of a similar one-bit register interconnection using the logically equivalent modules of Fig. 2(b). In this case, after each transfer the state is automatically restored to standard by the bistable flip-flop in each module.) The difference in the two cases is really this. In conventional dc systems, for example, the required gain characteristics for bistability are taken account of once and for all in the flip-flops, and the only requirement on the logic coupling circuitry is that it reliably trigger the flip-flop one way or the other. In magnetic systems, bistability is obtained by proper design of the entire loop from □→○→□. In this case, the specific form of the (flux) gain properties of the logic coupling loops is also important.

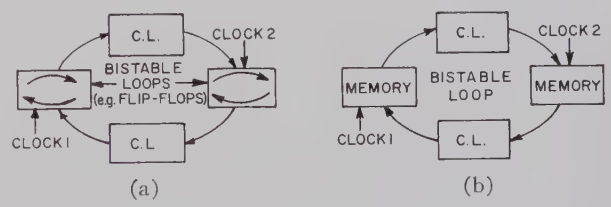


Fig. 8—Loop gain characteristics. (a) Conventional dc system—memory elements are inherently bistable and signals from logic networks are required only to "trigger" memory elements. (b) Conventional magnetic logic system. Bistability is a property of the entire loop including both memory and logic elements.

Practically, this difference in the source of the bistability is not important. In general, for any given design, it is merely necessary to set the allowed limits on the connecting logic circuits, and in each case do the best possible (that is, shortest lines) arrangement for least cross-talk, etc. Just how restrictive these limits may practically be, however, is another matter, and is discussed further in the next section. (In general, dc systems have considerable advantage in this regard.)

Finally, it may be noted that magnetic systems are possible in which the gain properties resemble those of Fig. 8(a). These can be obtained by use of different types

[4] A. J. Meyerhoff, Ed., "Digital Applications of Magnetic Devices," John Wiley and Sons, Inc., New York, N. Y.; 1960.

[5] E. Goto, "The parametron, a digital computing element which utilizes parametric oscillation," Proc. IRE, vol. 47, pp. 1304–1316; August, 1959.

of magnetic elements or different systems. For example, properly oriented thin-magnetic films that exhibit coherent rotation have only two stable states. Thus, it is only necessary to "trigger" the magnetization one way or another from its critical position, and the film will continue switching in that direction "on its own" until the stable state is reached.[6]

Comparison of Present All-Magnetic System to "Conventional" Core-Diode Systems

Although the most common computing systems realized with core-diode schemes are generally referred to as "magnetic computers," it may be helpful to make the following distinction. It is probably more accurate to think of these systems as diode-logic systems with the magnetic cores used for power amplification. The term "magnetic computer" is more reasonably applied to all-magnetic systems in which the magnetic elements (with the interconnecting wire) must do the whole job.

Magnetic systems, in general, are at a disadvantage with respect to conventional dc logic systems, or, as a matter of fact, compared to any systems that operate on voltage rather than current. Thus, in magnetic circuits, the coupling loops must carry current in order to provide suitable mmf's at the receiver. Any impedance in the loops therefore accounts for loss, which may be severe. In voltage-operated systems, impedance in the lines generally accounts only for delay, or slowing of the operation. Thus, in dc systems there is generally greater freedom to run "long lines." (In ac systems, line length may be more important in terms of phase shift than attenuation.) This may not be too severe in core-diode systems where the available gain is relatively high, but it becomes of more concern in all-magnetic systems where the available gain is generally lower.

Actually, in core-diode systems the diodes in the loops generally present more of a problem, as far as loss is concerned, than does the parasitic line impedance. To minimize this loss, the coupling-loop windings generally have relatively large numbers of turns (perhaps on the order of 10 to 100 turns, or so) in order to be able to operate with low currents in the coupling loops, in which case, the diodes do not offer high voltage drops. Turns are expensive, however, and as a matter of fact, one of the significant advantages of diodeless systems is that very low numbers of turns may be used. In the present case, almost all windings are *single* turn. Therefore, relatively high currents flow in the coupling loops, and a greater penalty is paid in regard to parasitic loop impedance. In fact, as indicated in Part I of this paper, in order to prevent excessive losses in the coupling loops under these circumstances, low-impedance strip lines (sticky tape sandwiched between copper shim stock) was used for constructing the coupling loops.

Finally, in a dc system the combinational-logic can be, and usually is, a cascade of logic modules (typically referred to, for example, as multi-level gating) by which relatively complex logic functions can be formed during a single clock time. This could also be done in core-diode systems, but generally leads to design difficulties. In the present all-magnetic system, this cannot be done at all. Rather, during each Advance phase only a straight OR function can be formed. This implies that it takes time to generate complex logic expressions. The small fan-in and fan-out in the present system even accentuates the time required to form functions. This appears to be one of the more difficult bridges to gap between this type of logic scheme and the more conventional logic realization schemes. Once a designer is accustomed to working in this environment, however, he should encounter no particular difficulty. It is merely necessary to know when a certain function is required, and then to determine when in the timing cycle it is necessary to start combining the appropriate variables so that the function is available at the proper time.

Specific System Logic

The material from here on is specifically related to the system under consideration. It is not intended to follow specific machine operations in detail (although from the material presented the interested reader should be able to synthesize the details of operation). Rather, some details on the major sections of the machine are presented in order to indicate the form of the logic synthesis.

The schematic of the machine is indicated in Fig. 9. It may be well to point out here that some of the logic arrangements may look somewhat strange, reflecting the fact that during the design period there were some "oscillations" of approach, with no final attempt to unify the design. The nature of these oscillations is indicated in the following section on the Cycle Timer.

To aid in following the logic, notice that the "logic section" of the schematic is laid out according to a specific plan. Thus, along any horizontal line, only modules of the same type appear—that is, all □'s or all ○'s. Further, the modules are arranged vertically for simplest visualization of the logical synthesis, that is, "time runs vertically."

The system is designed to perform (decimal) addition, subtraction, and multiplication. Since addition and subtraction are essentially variations on multiplication—*i.e.* multiplication by *one*—attention is paid here only to the necessary register arrangement for multiplication.

To multiply an n-digit number by another n-digit number requires a product (abbreviated PROD) register of $2n$-digit capacity. Such a register is indicated sche-

[6] D. C. Engelbart, "High Speed Components for Digital Computers," Wright Air Dev. Ctr., Wright-Patterson AFB, Dayton, Ohio, pts. A of Quart. Repts. 3 and 4, Contract AF 33(616)-5804; 1959.

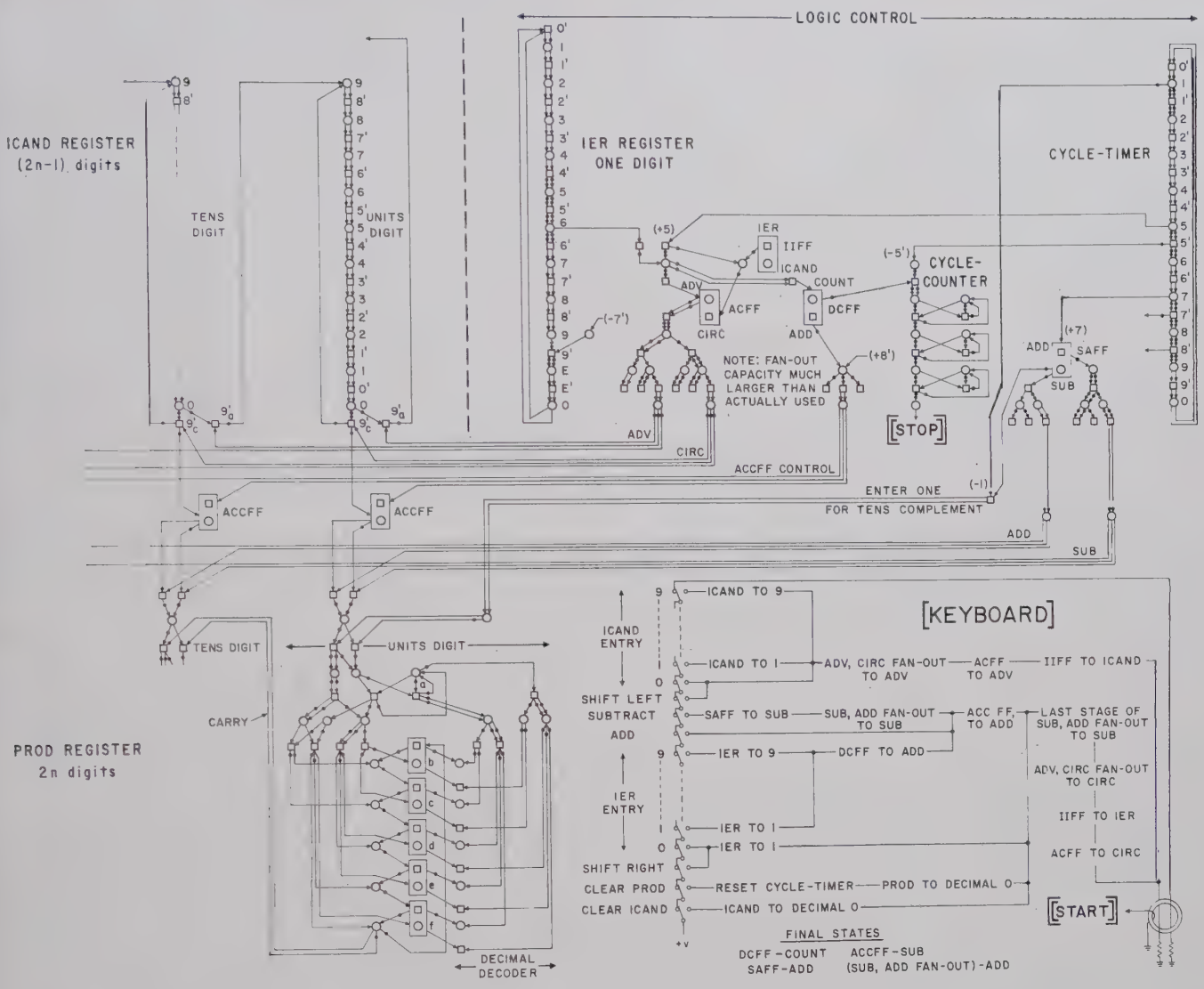


Fig. 9—Logic schematic; keyboard layout in lower-right section.

matically in Fig. 10(a) for the case $n=3$, or $2n=6$. Each digit of the PROD is really a decimal counter. The register is laid out with the most significant digit (MSD) to the left; therefore, carry signals propagate from right to left. The general multiplication problem, of course, is to multiply a multiplicand (abbreviated ICAND) by a multiplier (abbreviated IER) to obtain a PROD, as indicated in the numerical example of Fig. 10(b). In order to conserve on register capacity, the design calls for only a single IER digit to be handled at a time, so that the multiplication proceeds in steps, each IER digit being completely done with before the next IER digit is entered. This results in the need for only a single digit of IER capacity. Further, it was desired to enter the IER in the most natural manner, *i.e.*, MSD first. This requires an effective right shifting operation of the ICAND "over" the PROD. Thus, the ICAND is positioned with its least significant digit (LSD) in the central ICAND digit register, as indicated by the nu-

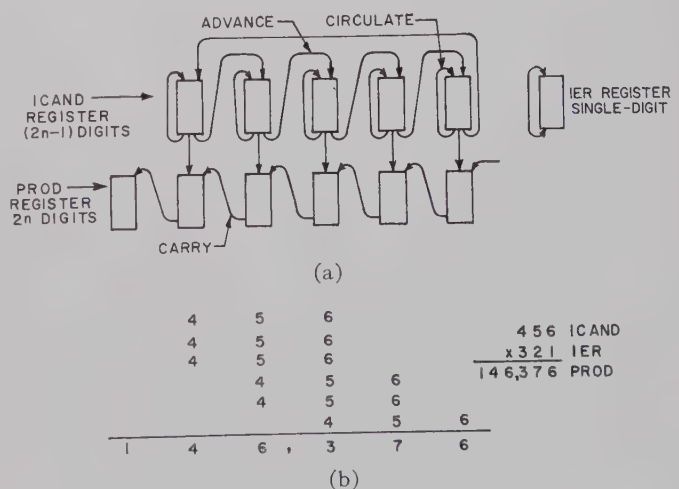


Fig. 10—(a) Layout of arithmetic registers. Multiplier handled a single digit at a time—most significant digit first. (b) Numerical example.

merical example. (For a full n-digit ICAND, the MSD, therefore, appears in the leftmost ICAND position.) To perform the indicated numerical multiplication, a *three* is entered into the IER register, with the consequence that the ICAND is added into the PROD three times and then shifted one digit position to the right. The next IER-2 digit results in the ICAND being added in twice more in its new position, and shifted again one position to the right. Finally, the ICAND is added in once more upon entry of the IER-1 digit, and the multiplication is complete, the answer being held in the PROD register.

Although the original intention was to design the machine for five-digit multiplication—*i.e.*, $n=5$—construction was halted after only three ICAND and three PROD registers were completed. Thus, two-digit multiplication can be handled, but for products greater than 999, the most significant product digit is not recorded. (Although it would have been more satisfying to watch larger numbers being handled, with consequent increase in indicator light activity, no significantly new information would have been obtained by merely extending the register capacity.)

Major Machine Sections

Cycle-Timer

This timer, indicated in the logic portion of the schematic, is a closed-loop, 10-bit shift register that is used to time all machine operations. Thus, the register contains only a single *one*, and at the beginning of any machine operation, it is insured (by external setting from the keyboard) that the *one* is in the ○ module labelled "0." Let us define a *clock cycle* as a single complete set of clocks. Hence, after one clock cycle, the contents of any register are shifted exactly one position, any bit in a ○ module being advanced to the very next ○ module. Therefore, it takes 10 clock cycles to advance the cycle-timer completely around once, or one timer cycle is equivalent to 10 clock cycles. As can be seen from the schematic, various logic circuits connect to different positions along the timer.

There are two points in particular to note about the register. First, the input to any module comes from only a single neighbor, and double transfer is used, although logically, double transfer is unnecessary. (It was noted in Part I of the paper that the $A \leftrightarrow B$ transfer operation—during which time the two elements of a module transmit to each other—is necessarily a lossy one in terms of flux gain. By using double transfer, however, there is no flux transfer necessary during the $A \leftrightarrow B$ clock, and again, in the spirit of conservation design, double transfer was used wherever possible.) Second, the series of transfers in the register appears to be an arbitrary arrangement of positives and negatives. This is a matter of history, and actually all transfers could just as well be positive, or negative, or any combination, as indicated in Fig. 4(c). (Initially, it was desired to make all transfers positive in order to reduce loading on the clocks. Thus, with all positive transfers and only a single *one* in the register there is only a single *one*-flux transfer per clock, whereas in an all-negative register there would be nine units of high-flux transfer on the ○→□ clock phase and only one unit of high-flux transfer on the □→○ phase. However, there was also some concern initially over the performance when three positive outputs were used. Thus, where a positive tap-off point was required, the inputs to that stage were made negative, the tap-off transfer was made negative, and the transfers to the adjacent register stage made negative; logically, this is identical to all positive transfers. Finally, there is some holdover in the design from a period in which it was felt that negative transfers, in general, were better than positive, so that negative transfers were favored wherever possible. As it turns out, however, all of these points are fairly insignificant, and actually any combination of transfers can be used with essentially equal performance.)

Multiplicand Register

The name for this register is abbreviated to ICAND register. Actually, it is a set of 10-bit registers which by controlled gating can be made to circulate independently (CIRC mode), or else can be connected into one long register (Advance, abbreviated ADV mode). Consider the 10-bit register labelled "Units Digit." The ○ modules of the register are labelled from "0" through "9," the labelling being significant in that a decimal number is entered into the register by setting to the *one* state the corresponding ○ module; thus, to enter a *seven*, the ○ module labelled "7" is set, all other ○ modules being cleared.

Let us now consider the gating. At the lower end of the register it can be seen that module "0" transmits to both modules $9_c'$ and $9_a'$ (subscripts "c" for CIRC and "a" for ADV). The logic control section determines which of these modules is "gated on" during each cycle of the cycle timer. The arrangement is sketched for convenience in Fig. 11. Whenever the set of CIRC input lines is in the *one* state, then the output of each $9_c'$ module is *zero*, regardless of the state of its associated "0" module. If, simultaneously, all ADV input lines are in the *zero* state, then each "9" module (top module of register) receives data directly from the "0" module of the register to its left—there being two negative transfers between these two modules. On the other hand, if CIRC = 0, and ADV = 1, then each "9" module receives data directly from the "0" module of the same register, and each register circulates independently. Either set of control lines is gated on for an entire cycle of the cycle-timer. Hence, during any one cycle, the entire contents of each register are advanced to the register to its right, or are circulated locally.

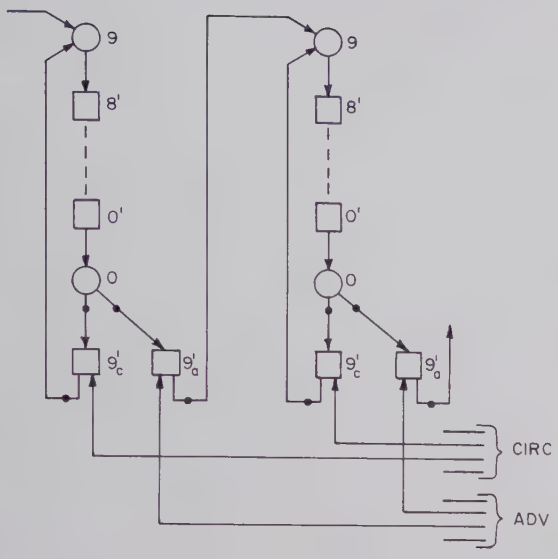


Fig. 11—CIRC and ADV gating in ICAND register.

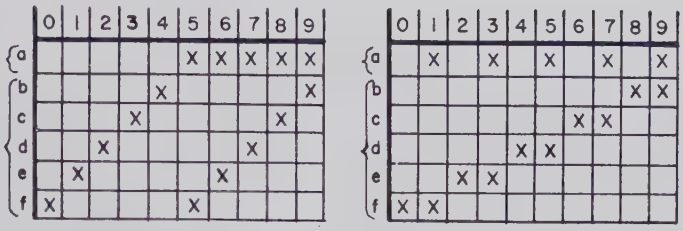


Fig. 12—(a) Bi-quinary representation. (b) Modified bi-quinary representation.

Product Register

Each PROD register has two sources of numerical input and one numerical output. One input source is the Carry signal from the adjacent less-significant stage. The other input is from the corresponding ICAND register. Each time that a *one* signal is received on either input line, the stored decimal count advances one. If both inputs are *one* simultaneously, the count advances by two. Whenever the count advances from nine to zero, a carry-out signal is issued to the next more-significant digit counter.

The PROD register is necessarily more complex than a simple shift register. This can be seen from the fact that the register must hold its state (count) even when no inputs are being received, although the clock generators may still be running. One possibility is to use a double-mode shift register like the ICAND register, except that here ten one-bit registers could be used to store the decimal count. At any instant, only a single such register is in the *one* state—all others are *zero*. Whenever the inputs are *zero*, the one-bit registers circulate independently and the count is held. When an input is received, the registers are effectively connected end-to-end, and the stored *one* advanced one position, to indicate an advance of one unit of the count. In the present case, because of the possibility of receiving two inputs simultaneously, there was an advantage to realizing the register based on a modified bi-quinary number scheme.

A conventional bi-quinary scheme uses six positions for decimal representation. The representation is indicated in Fig. 12(a). Assume that the number is stored in six flip-flops (labelled *a*, *b*, *c*, *d*, *e*, *f*). An *x* in a box represents a *one* state for the corresponding flip-flop. Thus, digit 6 is represented by flip-flop *a* and *e* in the *one* condition. Flip-flop *a* is in the *zero* condition for digits 0–4, and in the *one* condition for digits 5–9. A modified bi-quinary scheme representation is indicated in Fig. 12(b). Flip-flop *a* is in the *zero* condition for even digits 0, 2, 4, 6, 8 and in the *one* condition for odd digits 1, 3, 5, 7, 9.

The modified bi-quinary scheme is applied in the PROD register by having a "counter" stage [Fig. 6(b)] followed by five flip-flops connected in a ring. One and only one flip-flop of the ring is in the *one* condition at any time. The counter stage corresponds to the *a* element of the chart of Fig. 12(b). A *one* on the carry-in line or the ICAND input line causes the *a* element to count, *i.e.*, to change state. Whenever the *a* element goes from *one* to *zero* the ring of five flip-flops is caused to shift its contents a single position. Hence, the single stored *one* advances one position. As input pulses are received, the ten-digit representations are generated in sequence. The only special condition occurs if both input lines are *one* simultaneously. In this case, the *a* counter stage is bypassed, and the ring is directly caused to advance a single position.

The modified bi-quinary representation is easily decoded to a one-out-of-ten representation for straight decimal output representation. This decoder is included in the PROD circuitry, and involves the use of ten separate modules. The method of interconnection of these modules is clear from the schematic. Thus, if the lower flip-flop of the ring of five is in the *one* state, then we know that the count is "8," or "9"; interconnections from the *a* flip-flop determine whether the even or odd digit is actually correct. With the decoder arrangement shown in the schematic, for any particular count only a single decoder module remains in the *zero* state. (In the section on indicators in Part I of the paper, it was indicated that indicators could be connected so as to be excited when the module holds a *one* or a *zero*. In this case, the indicators are arranged to be excited for a *zero* state, so that for any count only a single indicator is excited.)

Multiplier Register

The multiplication scheme is designed to be carried out with only a single multiplier digit applied at a time. Any particular multiplier digit is therefore completely

used before the next digit is entered. Thus, only *one* decimal digit of IER storage is required. Note, however, that the register is eleven bits long. With this arrangement, every time the Cycle Timer completes one cycle, the data of the IER register "slips" one position. For example, with a decimal 4 digit initially inserted in the IER register (*i.e.*, a binary *one* in the position labelled "4") it takes four complete timer cycles for the binary *one* to work down to the "0" position. During these four cycles, the gating is arranged so that the ICAND registers circulate independently (CIRC mode) with their contents being added to the PROD. When the *one* of the IER register works down to decimal *zero* position, the gating then changes to control a single ADV cycle, during which time the ICAND register shifts one digit to the right. The PROD registers are not affected during this cycle. After the single ADV cycle, there are seven more non-add CIRC cycles during which time the ICAND registers circulate their data again, but without affecting the PROD registers. These extra seven cycles are provided merely for time delay to guarantee completion of any Carry propagation signals between the PROD registers. These cycles are counted by the Cycle Counter.

Cycle Counter

This counter is made up of three binary counters in cascade. Thus, it takes eight input signals to cause a signal out of the last stage. Every time the Cycle Timer completes a cycle, during the single ADV and the following seven CIRC cycles at the end of a multiplier process, the Cycle Counter receives a single count. By the end of the last CIRC cycle, a signal from the Cycle Counter causes the clock generators to stop. This Stop signal is so timed that all registers end up in proper alignment with respect to decimal digit representation. (Starting and stopping of the clock generators is considered in more detail in the next section.)

Actually, seven dummy CIRC cycles is somewhat more than necessary to cover the maximum required carry propagate time in the PROD register. However, the eight-cycle counter is convenient to use for the ICAND entry procedure also. (See ICAND entry under Keyboard Operations.)

Keyboard Operations

The keyboard is provided with a separate button for the initiation of each operation or "order." The basic keyboard layout is schematically indicated in the lower right portion of Fig. 9.

Each operation button is a single-pole, double-position switch. When a key is depressed, a current is caused to flow through an associated line which "sets up" the machine for the particular operation. As the key is released, a START signal initiates the clock program generator, which runs continually until a STOP command is received from the Cycle Counter. All keyboard-initiated operations start and stop the clock generators in exactly the same way. Note that *all* modules of the system experience exactly the same clock program. No "extra" equipment is used to gate the clock pulses to various portions of the machine at different times.

The clock generator is arranged so that when the clocks are started the very first pulse is Clear □. (As indicated in Part I of the paper, simultaneous with the Clear □ pulse is the $SS_{\bigcirc}$ pulse, which sets all of the synthetic output cores of the ○ modules.) When the clocks are shut off, the last pulse is the $(A \leftrightarrow B)_{\bigcirc}$.

For any operation, the signal for stopping the clocks comes from the output of the Cycle Counter. When that signal is obtained, the clocks stop at the end of the very clock cycle during which the signal was received. By terminating the clock cycle on $(A \leftrightarrow B)_{\bigcirc}$, just before Clear □, the maximum information is retained after the clocks are terminated; that is, the ○ and □ modules both contain data. Advantage is taken of this fact by placing indicator lights on both the ○ and □ modules—*e.g.*, in the PROD readout decoder.

During machine setup for any operation, it is necessary that all ○ modules are in the proper state, that is, *one* or *zero*. Note that for all ○ modules that are to be set, only the MAD's in the module need be switched, since on the very first clock pulse, $SS_{\bigcirc}$ will set all of the synthetic output cores of those modules.

The START signal is indicated as coming from a conventional magnetic core. This arrangement is used to prevent any complications from key-contact bounce. As a key is depressed, the current that flows in its associated machine set-up line returns to ground through a winding on the core. Thus, as any key is depressed the core is switched, but the resulting switching voltage in the START line is of the wrong polarity to start the generators. As the key is released, the core is switched back, and a proper signal on the START line initiates the program. The core switches completely in approximately one microsecond, and any bouncing of the key therefore, has no effect, *i.e.*, will not cause extra START signals. With this mode of operation, there is assurance that the machine is properly set up before the clocks begin.

Set-up of the machine for a particular operation, by depressing the associated control button, implies entering numerical data into registers (where appropriate) and switching control flip-flops into their appropriate starting state for the particular operation.

Although each operation key is indicated as a single pole switch, a second pole is used for ease in controlling the light-indicator generator, which is shut off during the execution of any machine operation. (Actually, the light generator could be left on permanently, even during machine operation, but this results in a small decrease in the allowed per cent range on the clock current amplitudes.)

ICAND Entry

A single array of ten pushbuttons is used for the ICAND entry, which proceeds with the MSD (most significant digit) first. [It was indicated in connection with Fig. 9(a), that the ICAND is positioned with the LSD in the central register of the $2n-1$ registers.] As any button is depressed it enters a binary *one* into the corresponding position of the register just to the right of the central ICAND register. Thus, decimal 7 is represented by a binary *one* in the seventh position of the register. Before the next less significant digit is entered into the same ICAND register, the last-entered digit must be shifted one position to the left towards the MSD end. Since the ICAND registers are not arranged for left shifting, however, the left shift of one digit position is achieved by an equivalent right shift. In the initial design for five-digit numbers, *i.e.*, $n=5$, the required right shift was equal to $2n-2$, or 8 shifts. The eight-count Cycle Counter was ideally suited to time this operation. For the present machine, in which $n=2$, the desired number of shifts is two; however, any multiple of two is also acceptable. Thus, for this operation the initial design of eight shifts was preserved, the only effect being that the entered number circulates three times around the ICAND register rather than once.

Thus, when an ICAND key is depressed, a binary *one* is entered into the appropriate position of the central ICAND register, and the machine is set up so that when the clock generator starts, upon release of the key, the machine programs eight non-add ADV cycles.

IER Entry

Multiplication is accomplished by: 1) clearing the PROD register (or not, if the new product is to be added to a previous result), 2) entering the ICAND, and 3) entering the multiplier (abbreviated IER) most significant digit first. A single array of ten pushbuttons is used for IER entry. As each IER digit is entered, the ICAND is added to the PROD a number of times indicated by the value of the IER digit, and then advanced one digit position to the right, to be ready for entry of the next less significant IER digit. This process is continued for as many IER digits as desired. At each step the PROD holds the correct partial product to that point.

Depression of an IER key enters a binary *one* into the appropriate position of the IER register and sets up the machine so that with the start of the clock generator, upon release of the key, the multiply program is carried out. The nature of this multiply program has already been indicated under "IER register" in the previous section.

Add, Subtract

Adding is essentially the same as multiplying by unity (*i.e.*, IER = 1). The main difference is that the one ADV cycle normally following each IER entry operation is converted to a CIRC cycle. Thus, the number in the ICAND register (which now plays the role of the *addend*) is added once to the PROD register (which now plays the role of the SUM register) but ends up in exactly the same position. To add the same number again, the ADD key is depressed a second time.

Depression of the ADD key sets up the machine so that after the single CIRC cycle, during which the ICAND (Addend) is added once to the PROD (Sum), the machine programs eight non-add CIRC cycles, rather than one ADV followed by seven CIRC cycles, as after an IER entry operation.

Subtraction is identical to addition except that the tens complement of the ICAND is added to the PROD during the first CIRC cycle.

Shift ICAND Left, Right

The shift-ICAND-left operation is identical to an ICAND entry except that no connection is made to the actual ICAND register. Thus, this operation calls for the right shift of eight places (equivalent to one left shift) without disturbing the numerical data of the ICAND registers.

The shift-ICAND-right operation is identical to the operation of multiplying by IER = 0. The PROD is not disturbed and the ICAND ends up shifted one place to the right.

Clear ICAND; Clear PROD

As their names imply, Clear ICAND (PROD) results in decimal zero being inserted into the ICAND (PROD) register. Although these operations could be performed directly on the respective registers without starting the clock generators, for the following reasons it was simpler, in particular for Clear PROD, to actually initiate a machine program. Recall that for proper machine set-up for any operation, it is necessary to insure that all ○ modules are in their proper state. To properly put the PROD into a decimal zero state means actively setting, or resetting, ○ modules in each PROD register, other than those directly associated with the flip-flops. It turned out to be simpler to set merely all flip-flops from the keyboard, and then have the machine run through eight non-add CIRC cycles, after which all of the remaining ○ modules are then properly set for the decimal zero condition.

Conclusions

Some implications of this study have been indicated in the Conclusions of Part I. A few additional thoughts are indicated below.

The logical synthesis techniques suitable for handling the all-magnetic logic module used in this study are similar to those that would be used in core-diode logic systems in which only a single level of diode logic was

permitted during each advance clock phase. In the present case, however, the permitted fan-in and fan-out is probably smaller than in most core-diode systems, where there is generally considerably more gain per stage possible. The facility for only single-level logic, coupled with the small fan-in and fan-out in the present module, means that complex logic functions must be generated slowly, step-by-step. Thus, once it is determined when, in the process, a certain function is required, it is necessary to start combining logic variables sufficiently in advance so that the function synthesis is completed in time. This step-by-step generation of functions results in the need for more modules than might otherwise be needed if larger fan-in were possible, for instance, but aside from that, one soon gets used to anticipating such function generation, and the synthesis techniques are not particularly different from customary logical design.

One very significant property afforded by the use of multiaperture elements is nondestructive read-out. It has been indicated how such elements can be used directly to control indicators (such as incandescent bulbs) in such a manner as to be nondestructive to the information state of the element. Thus, with the MAD elements used in this system, a few hundred milliwatts of power, per module, may be directly controlled in this manner. This is of interest not only for indicating arithmetic results, such as from registers, but it also affords a general means for indicating logic control states. Of course, other external elements aside from incandescent bulbs may be controlled directly from the MAD elements in this manner, without the need for matching, or interface, devices. Thus, it may be possible to control valves, voice channels, etc.

The above discussion relates to nondestructive read-out to some sort of output device. However, the ability to transmit nondestructively from one logic module to another may also lead to interesting logic structures. For example, in some all-magnetic schemes it is possible to clear an element as well as set it from an input loop. Thus, it may be possible to use a single element as a logical flip-flop, with its state continually read out in a nondestructive manner. In this sense, the logical structure would much more closely resemble the structure of logic systems based on conventional flip-flops. This is suggested merely as an example of the potentially increased power afforded by multiaperture devices over conventional magnetic toroids.

Finally, in all-magnetic systems, multiaperture devices may be more interesting than conventional magnetic toroids in another manner as well. With respect to complexity, at one end of the spectrum are those systems built with simple devices (toroids) and relatively complex interconnections. At the other extreme, is a single piece of material in which is synthesized an entire computer, with all flux transfers taking place within magnetic paths, no coupling loops required at all. If one plotted some measure of simplicity over this spectrum—a measure including costs of device development, ease of design, performance, etc.—it is the opinion of the author that the function would peak at a point away from either extreme. The single-device computer does not even appear to be a practical goal (at least with bulk ferrites as the basic medium). On the other hand, although there are advocates of the thesis arising from the question, "Why build complex devices when you can do everything with simple toroids?" it is felt that the present system is simpler than a comparable one made purely with simple toroids. Furthermore, it is felt that even more function may be profitably incorporated into the multiaperture devices. For example, it has been indicated that single devices which can perform logical positive or negative transfer can be realized.[7,8,9] Further, circuit functions like clipping, which in this system were realized by the use of separate toroids, could also easily be incorporated into the devices themselves.[8]

Acknowledgment

The author wishes to thank the Air Force Cambridge Research Laboratories, Bedford, Mass., for permission to publish this paper.

[7] H. D. Crane, "Magnetic core logic element," Patent No. 2,935,622; June 12, 1958.
[8] D. C. Engelbart, unpublished notes, Stanford Res. Inst., Menlo Park, Calif.
[9] D. R. Bennion, "New multiaperture magnetic logic element," *J. Appl. Phys.*, vol. 31, suppl., pp. 129S–130S; May, 1960.

A 2.18-Microsecond Megabit Core Storage Unit*

C. A. ALLEN†, MEMBER, IRE, G. D. BRUCE†, MEMBER, IRE, AND E. D. COUNCILL†, MEMBER, IRE

Summary—A magnetic core memory is described which has a read-write cycle time of 2.18 μsec, an access time of 1 μsec, and a storage capacity of 1,179,648 bits. The array configuration and the design of the driving system are shown. The core and transistor requirements are discussed, and a description is given of the sensing and the driving circuitry. Design factors which governed the choice of the 3-dimensional system organization are presented.

Introduction

THE IBM 7302 Core Storage is a large magnetic core memory with a cycle time of 2.18 μsec (Fig. 1). It uses ferrite toroids and is a coincident-current, three-dimensional system; 16,384 words of 72 bits each are addressable by random access in a single array of 1,179,648 cores.

Fig. 1—The IBM 7302 core storage unit.

The development of this memory was originated as part of a program for high-speed memories for the IBM STRETCH Computer. The objectives for the medium-speed memory for this computer have been met,[1] but with twice the storage capacity that originally was set.

This basic memory, with logic modifications, now is used in the IBM 7090 and the IBM 7080 computers, in addition to the STRETCH computer. The first memory was shipped to a customer as part of a 7090 system in November, 1959, operating in a 2.8-μsec cycle with an access time of 1 μsec.

The memory is all solid state. It uses graded-base transistors for logic and sensing and the IBM medium-power graded-base transistor as a core driver. There are no programming restrictions so that any random address may be selected repetitively at full cycle rate.

Array Design

Primary considerations in selecting the core type for this memory were core switching time and drive requirements. In order to obtain the desired speed in an all solid-state machine, maximum core switching time was set at 0.4 μsec with nominal half-select current not to exceed 0.6 ampere at 0.1 μsec rise time.[2,3]

The pertinent core parameters are

1) Physical size = 30×50×12 mils
2) Full-select current = 1.17 amperes nominal with T_r = 0.1 μsec
3) Delta noise = 1.0 mv per pair.

Under marginal drive conditions:

4) Switching time = 0.43 μsec maximum
5) Minimum "one" = 110 mv
6) Maximum "zero" = 33 mv.

Each digit plane consists of 16,384 cores wired into a 128×128 matrix. Center-to-center core spacing is 0.0625 inch, and the over-all plane size is approximately 10 inches square. Fig. 2(a) shows a core array, and Fig. 2(b) shows a completed plane.

Four separate sense and inhibit (z) windings are used in an interlocking arrangement. Each winding is rectangular and links a 32×128 matrix. The sense and inhibit segments are oriented at right angles to each other, which results in 1024 cores which share the same inhibit and sense winding. The primary advantage of this arrangement is the reduction of core noise on a sense segment—noise that results from pulsing an inhibit segment.

All windings are conventional with the exception of the sense winding. Fig. 2(c) shows the sense configura-

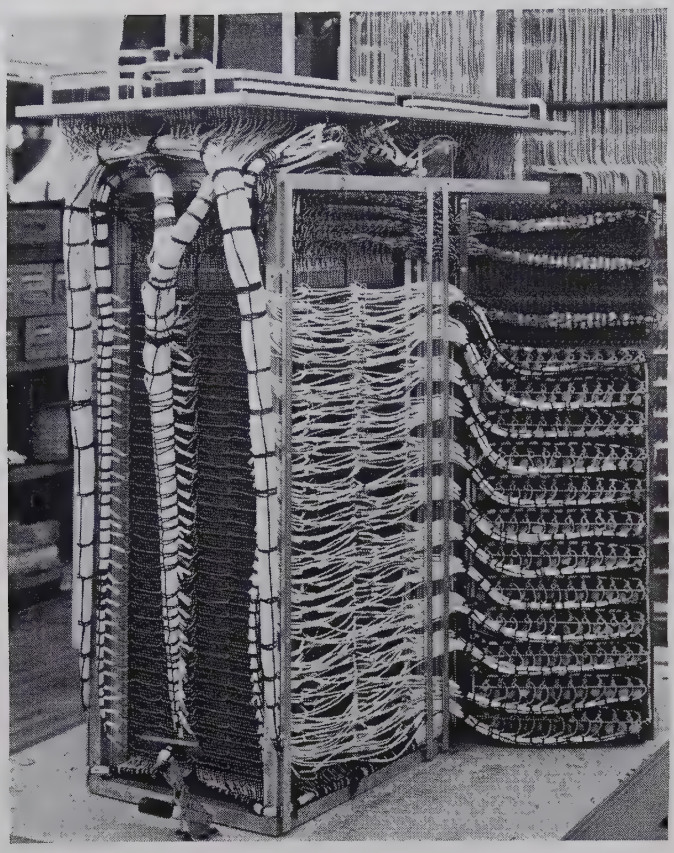
* Received by the PGEC, October 22, 1960; revised manuscript received, December 22, 1960.

† Data Systems Div., IBM Corp., Poughkeepsie, N. Y.

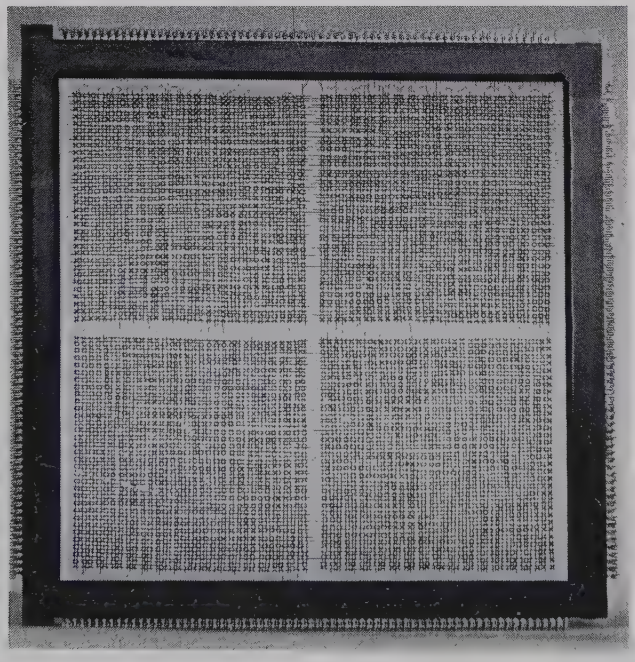
[1] S. W. Dunwell, "Design objectives for the IBM STRETCH computer," *Proc. EJCC*, pp. 20–22; 1956.

[2] M. H. Cook, E. H. Melan, and E. C. Schuenzel, "High-speed ferrite core for a coincident-current memory," presented at Electronic Components Conference, Philadelphia, Pa.; May, 1959.

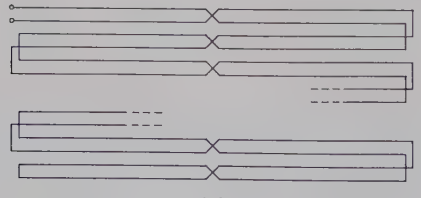
[3] B. R. Eichbaum, "The development of high-speed coincident-current memory cores," presented at Conference on Magnetism and Magnetic Materials, Detroit, Mich.; November, 1959.



(a)



(b)



(c)

Fig. 2—(a) A core array. (b) A completed core plane. (c) Sense-winding configuration.

tion which has the following advantages over the familiar diagonal sense winding:

1) Shorter physical and electrical length result in decrease in delay and attenuation of signal.
2) Noise cancellation is improved by virtue of smaller time delays between cancelling components.
3) Because the sense is now parallel to the X drive line, the transmission line properties of the X lines become approximately equal to the Y lines, which are parallel to the inhibit windings.

The rectangular shape of the sense segment was chosen as optimum for use with staggered read. This shape provides a 50 per cent reduction in core half-select noise at strobe time, in comparison with a more conventional square segment.

The wire orientation with the core is maintained throughout the plane. In order to reduce and provide uniformly-distributed capacitance between the sense and X lines, it was necessary to insure that these lines were separated by the inhibit and Y lines. Considerable electrostatic coupling between inhibit and Y lines resulted from this choice, since inhibit and Y were then immediately adjacent and parallel. Careful pulse timing was required to avoid trouble caused by the propagation of capacitively-coupled noise spikes on the Y lines common to an inhibit segment, when a large number of inhibit drivers were turned on.

The word-selection lines of 72 planes are connected to form a 16,384-word array of 72 bits per word. For 7090 operation this array becomes 32,768 words of 36 bits each, with simultaneous read-out of two words by logical modification.

The relatively long word selection line length (about 72 feet), fast rate of change of drive current, and close spacing between adjacent lines indicated a serious problem area. Electromagnetic and electrostatic coupling between adjacent lines was great enough to produce noise spikes on unselected lines of greater than 35 per cent of the drive current during drive-pulse rise and fall.

A transposition technique was used in order to combat this cross-talk problem. The external-array jumpers for both the X lines and the Y lines were arranged so that, except at only a few planes, no two lines were immediately adjacent in the stack. By transposing, or cross-jumpering, the adjacent drive lines every few planes, the worst coupling became that between lines no closer than two positions of line spacing. A further step toward reduction of electromagnetic coupling was an array layout which called for alternate planes to be turned over so that approximately 0.060 inch separated core mats in a pair of planes. The spacing between pairs of planes was 0.75 inch, accounted for solely by the thickness of two plane frames. The combined effects of this array layout and wiring were to reduce the adjacent line crosstalk to less than 10 per cent of the drive current. This was a satisfactory solution, inasmuch as no problems of drive-line crosstalk were encountered in the final machine.

All windings approximate lossless transmission lines and are terminated in pure resistances equal to their characteristic impedance. The X and Y lines have practically identical properties, since both lines have the same distributed inductance and very nearly the same distributed capacitance.

The line characteristics in air were:

	Z_0 (ohm)	T_d (nsec)
X, Y	150	150
Z	120	60
Sense	185	52

The requirement of continuous selection of one address necessitated using some means of core temperature stabilization to prevent undesirable changes in core properties as a result of heating. With a power dissipation of about 42 mw in each switching core, it became necessary to provide a liquid environment for the array. This choice resulted in good heat transfer away from the heated core and uniform temperature distribution in the array.

The liquid selected was a commercial inhibited transformer oil. The increase in the dielectric constant of the array environment caused changes in both Z_0 and T_d of all lines.

The line characteristics in oil were:

	Z_0 (ohm)	T_d (nsec)
X, Y	100	220
Z	90	80
Sense	130	65.

The undesirable increases in time delay made low dielectric constant a most important consideration in selecting a temperature stabilization medium. Some fringe benefits were derived from the oil-cooled system. Lower rated power resistors could be used than would be possible in air. The tank was more compact than a blower system would have been. The reduction in characteristic impedance materially eased the inhibit driver requirements.

The entire core array, load-sharing switch assemblies, and terminating resistors were immersed in a tank containing commercial inhibited transformer oil.

Core Drive System

The drive system for the memory requires driving nominal half-select currents of 0.585 ampere into a terminated transmission line. With oil immersant, the characteristic impedance of the X-Y lines is 100 ohms, requiring read and write pulses of 58.5 volts. Inhibit drive requires 53 volts nominally into a characteristic impedance of 90 ohms.

The X-Y drive lines must be driven bidirectionally for read and write currents of opposite polarity. This imposes a voltage requirement of at least 125 volts on the peak inverse voltage if a direct-drive system is employed. This is beyond the limit of the breakdown voltage that can be withstood reliably by any existing high-speed core-driver transistor. A transformer drive is dictated. The high inductance of a conventional matrix switch, coupled with the increased current needed to provide voltage step-up, makes the conventional switch unattractive. A load-sharing matrix switch,[4] however, permits voltage step-up without high-series inductance and without excessive power per driver. The X-Y drive system uses a load-sharing matrix switch to provide read and write pulses with reasonable transistor dissipation.

Eight 16-output load-sharing switches are used for each address selection coordinate of the memory, for a total of sixteen switches. Each switch (Fig. 3) has 32 single-turn input windings and 16 nine-turn output windings plus a bias winding. The decoding logic, from the memory address register, selects an address by generating a pattern for turning on drivers on one switch of the X-axis and a pattern for one switch of the Y-axis. In each of the two switches, 16 drivers turn on at read-time and the complementary 16 drivers then turn on at write-time. Drivers are selected in pairs, and the address determines which driver in each pair turns on for read and which one turns on for write. One core in a 16-output switch then receives all the current from 16 drivers in one polarity for read and all the current from the complementary 16 drivers in the other polarity for write. The currents through the 15 unselected cores cancel, so no net drive is applied to these cores. The dc bias winding provides a slight average current unbalance in each switch core, so that the maximum switch-core flux will be available since the core is in the low remanent state at the start of each read-current pulse

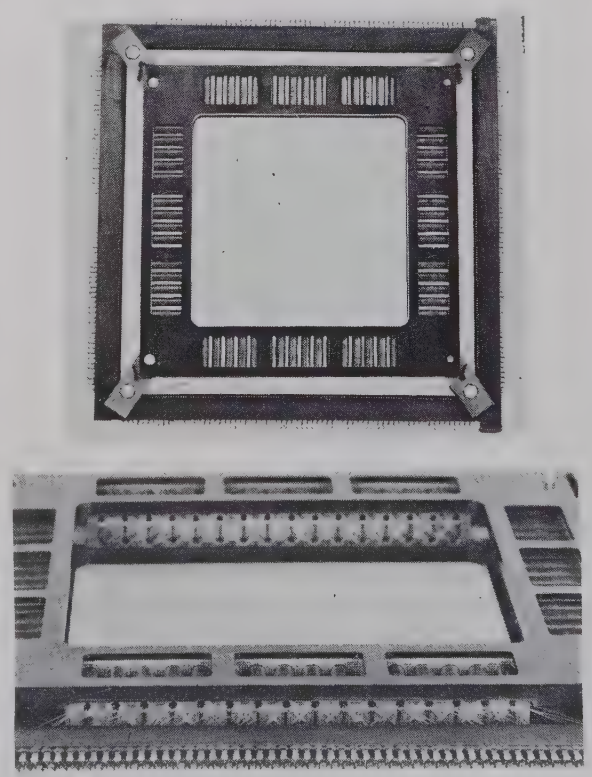


Fig. 3—Load-sharing matrix switch.

[4] G. Constantine, Jr., "A load-sharing matrix switch," *IBM J. Res. & Dev.*, vol. 2, pp. 204–211; July, 1958.

The switch-voltage step-up ratio of 9:1 requires less than 8 volts across each of the single-turn primary windings. The nominal current for each X-Y driver is 56 per cent of the half-select current plus one-sixteenth of the switch-core magnetizing current and bias current, or about 0.360 ampere. The total resistance of the leads plus switch windings and a 56-ohm resistance to +30 volts provide the resistive termination for the X-Y switch drivers.

The X-Y driver circuit uses the IBM medium-power graded-base unit as the output transistor. This transistor is a germanium unit packaged in an air-cooled sink. It is operated about 3 volts out of saturation, with a zener diode in a feedback loop determining the output operating level.

The inhibit current is supplied by a direct drive to the 128×32 plane segment. The Z winding is terminated in a 90-ohm resistor plus wiring resistance to a +60-volt supply. A clamping diode is used to prevent excessive peak voltage from being applied to the driver transistor.

The Z driver circuit also uses the medium-power transistor in the circuit shown in Fig. 4. The X-Y driver circuit and the Z circuit are basically the same configuration. Included in the Z-driver transistor specifications are a grounded-emitter large signal-current gain greater than 100 at 0.660 ampere and a collector-to-emitter breakdown voltage rating of 90 volts.

The Z-driver timing is staggered for two reasons:

1) First, the write-pulse propagation time is 220 nsec through 72 planes. The inhibit-pulse timing must overlap the write pulse in each plane. The total memory cycle can be reduced, however, by time-staggering the Z pulses to overlap an early write pulse in the first planes and to overlap a delayed write pulse in the last planes. Z-pulse width can be reduced, and read time can start into the top plane sooner for the next cycle of operation.
2) Second, the Z windings are parallel to the Y-axis windings in each segment. When the Z pulse turns on and off, charging currents are induced on the Y windings. The freedom to adjust the Z-pulse timing to groups of planes allows some reduction in the cumulative effect of these charging currents by phasing.

Both the rise time and the fall time of drive current into the array are on the order of 100 nsec. The array X-Y lines present a characteristic impedance to the drive of about 100 ohms in oil. The delay of the X and Y currents is 200 nsec, from input to termination. The Z-current delay is on the order of 80 nsec with a 90-ohm characteristic impedance. Each line must be terminated in a resistance equal to its characteristic impedance to avoid reflections and long recovery times. A drive line, terminated in a matched impedance, thus presents an essentially constant resistive load to the drive system.

Some slight distortion of the drive current occurs due to wire resistance (Fig. 5). The current into the first plane decreases slightly as the series resistance of the line becomes effective. During the initial part of the pulse, the line presents a load of 100 ohms, but this increases as the wavefront propagates through the array. The peak current out of the terminal plane is, therefore, lower than the peak input current for the first plane. The design of the load-sharing matrix switch with a series resistance in each primary driver allows some primary winding voltage change. This helps compensate for wire resistance and improves the matching of the drive system to the array.

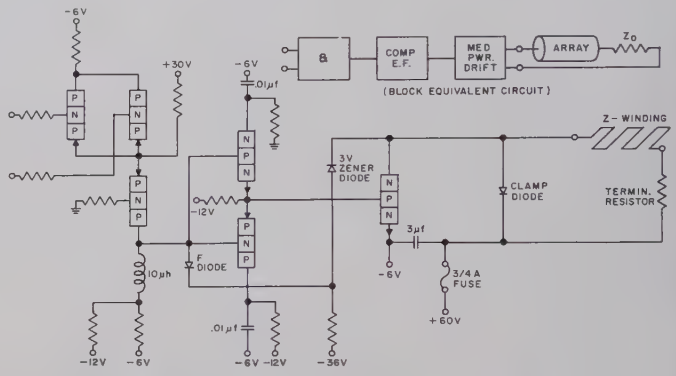


Fig. 4—Z-driver circuit schematic.

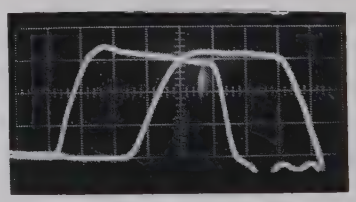


Fig. 5—X-read current pulses into and out of array. Scales: 100 nsec/cm; 200 ma/cm; 1 large division/cm.

Sensing

The segmented sense winding is used for the following reasons:

1) It reduces the delta noise since the Y-axis drive half-selects only 32 cores on each sense segment. Delta noise at read-time from the rise of current of the second of two staggered read pulses is only due to 15 cancelling pairs of cores. Total noise on the sense line due to read drive is then the result of the noise from these 15 core pairs, the half-select noise of one uncancelled core, and the noise level remaining from the first of the staggered read-drives on the X-axis which intersects 128 cores in each plane and is parallel to the sense winding.
2) It decreases the time difference between the sense signals traveling in two different directions away from a switching core in the plane. If this difference is too great, two signals of half-amplitude will be sensed at different times. Also, strobe timing

needs to be wider and without segmenting some increase in cycle time results.

3) It reduces the common-mode Z noise on each sense amplifier to one-sixteenth the noise of a full 128 $\times$128 plane.

The sense amplifier is shown in Fig. 6. Four identical preamplifier channels are used for each plane. Each of the four sense segments in a plane is sensed by a separate preamplifier. The outputs of these four preamplifiers are then mixed so that a "one" signal on any channel is amplified in the final stage to set the data register. The amplifier uses the usual time and amplitude discrimination by strobing and setting a clipping level. In addition, frequency discrimination is used. The circuit cutoff frequency is designed to block much of the relatively high-frequency noise on the sense winding.

Fig. 7 shows a typical sense output. Noise prior to strobe time is attenuated in the sense amplifier by frequency discrimination more than the "1" signal. The "1" signal has a relatively high amplitude for a band of frequencies in the range of 200 kc to 1.25 Mc. High cutoff frequency of the preamplifier stage is about 2 Mc. Low cutoff frequency is roughly 100 kc due to the interstage transformer.

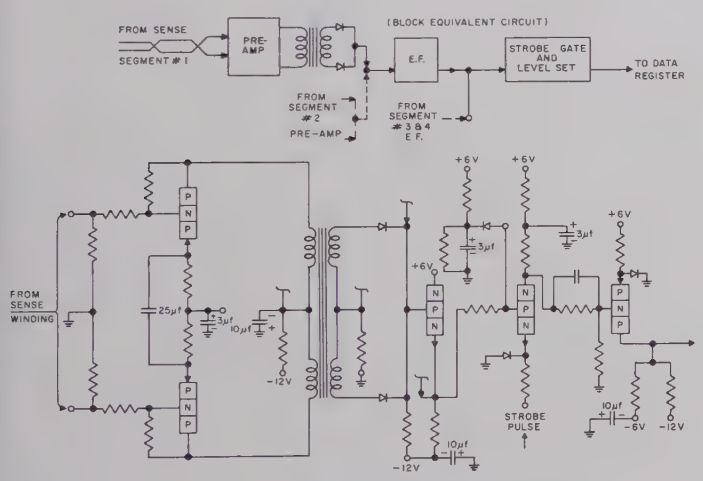


Fig. 6—Sense-amplifier circuit schematic.

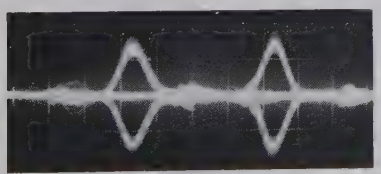


Fig. 7—Sense-segment output for all "1's" with memory cycling through all addresses. Scales: 500 nsec/cm; 100 mv/cm; 1 large division/cm.

Sense-amplifier strobe timing is staggered in groups of planes for reasons similar to those for the staggered Z timings. As the read pulse propagates through the 72-plane stack, the sense strobe timing is adjusted to coincide with the core peaking time for each plane.

Fig. 8 shows the cycle timing for this memory.

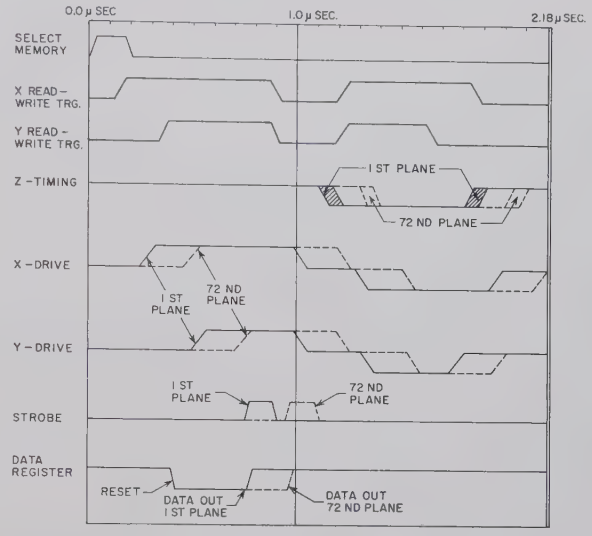


Fig. 8—7302 core storage timing.

Conclusions

The IBM 7302 Core Storage is six times faster than its predecessor, the 738. Although the storage capacity is the same for both machines, the cost of a 7302 is similar to that of a 738. Thus, a significant improvement in speed has been achieved at low cost with a smaller physical size.

There is every indication that even faster magnetic core stores of equal or larger capacity will become technically feasible. Larger and faster three-dimensional memories can be achieved with the aid of improved ferrites, improved components and a rapidly increasing memory technology.

Acknowledgment

Many people in IBM have contributed to the components and system design of this memory. The contributions of Magnetic Device Development, for the core, and Semiconductor Development, for the driver transistor, are particularly noteworthy. The success of the final design and production of this memory was due to the efforts of the members of the Standard Memory Development Group. Special credit must be given to S. W. Dunwell for establishing the ambitious goals of the STRETCH Computer, of which this was a part.

Matrix Switch and Drive System for a Low-Cost Magnetic-Core Memory*

WARREN A. CHRISTOPHERSON†, MEMBER, IRE

Summary—A unique system of ferrite-core matrix switches and drivers has been developed for a low-cost magnetic-core memory. The memory uses coincident-current techniques and has a capacity of 10,000 characters with seven bits per character. A 20-μsec read-compute-write cycle features serial-by-character processing. Approximately 7 μsec is computing time, and 13 μsec is read-write time.

The matrix switch requires only two sets of five drivers to select one out of 100 individual outputs. The drivers operate in an unusual three-out-of-five coding arrangement. A Set and Reset a driver, each using four transistors in parallel, are also required for the matrix switch. Two matrix switches provide the 200 X-Y half-select drives for a 100×100 seven-plane core array. At read time, two half-select current pulses of 250 ma to 300 ma are emitted with an effective 10 per cent to 90 per cent rise time of 0.3 μsec. At write time, half-select current pulses with 1.2-μsec rise time are emitted on the same selected lines, but in the opposite direction.

A new method of timing the drive current allows the read pulse to rise in 0.3 μsec, even with a low-voltage power supply and an inductive load that would otherwise limit the rise time to 0.8 μsec.

All current-driving circuits use alloy junction transistors. The drive current is furnished from a 10 to 12 volt power supply, and temperature compensation of the drive currents is accomplished through control of power-supply voltage. Operating temperatures range from 10°C to 40°C.

Introduction

THIS PAPER describes the organization, design, and operation of a ferrite-core matrix switch and drive system developed for a low-cost magnetic-core memory. Although core memories with matrix switches are well-known throughout the industry,[1–3] this development is believed to be significant in its achievement of a reliable, simple, low-cost, and compact core storage system utilizing current technology.

Cost reduction without sacrifice of function was the primary objective; hence, specifications for size and speed were conservative, *i.e.*, a capacity of 10,000 characters and a serial processing cycle of 20 μsec.

The development and successful operation of a complete memory requires consideration of many functions. Major emphasis in this paper, however, is placed on the more unique features and design of the matrix switch and drive circuits.

* Received by PGEC, October 25, 1960; revised manuscript received, February 15, 1961.

† General Products Div., Development Lab., IBM Corp., San Jose, Calif.

[1] J. W. Forrester, "Digital information storage in three dimensions using magnetic cores," *J. Appl. Phys.*, vol. 22, pp. 44–48; January, 1951.

[2] J. A. Rajchman, "Static magnetic matrix memory and switching circuits," *RCA Rev.*, vol. 13, pp. 183–201; June, 1952.

[3] W. L. Morgan, "Bibliography of digital magnetic circuits and materials," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-8, pp. 148–158; June, 1959.

Description of the Memory System

The memory system block diagram, Fig. 1, illustrates the application of the matrix switch and drive circuits. The 10,000-character core array is made of seven 100×100 memory-core planes identical to those used in the IBM 7070 and IBM 1620 systems. This seven-plane memory is operated in the standard coincident-current mode. The matrix switch and drive circuits are required to decode a four-digit binary-coded decimal (BCD) address from the address register, select the corresponding switch cores, "one" of 100, in each of two matrices, and emit a half-select current pulse on the corresponding X and Y half-select lines. Each half-select line drives one line of 100 memory cores in each of the seven planes in series. Thus, seven memory cores, one in each memory plane at corresponding X-Y line intersections (the four-digit address), read out to the sense amplifiers. The sense output is set into a seven-bit Read Register and the data processing, or compute, portion of the cycle is performed.

Following compute time, the pair of matrix switches generates two half-select "write" pulses on the two lines where "read" pulses were emitted. Thus, writing always occurs at the last address that has been read. The Z drivers, gated by a Write Register holding the result of the compute operation, are able to inhibit the writing action in a given memory plane. This permits the writing of a "one," or inhibits the writing of a "one," which is equivalent to the writing of a "zero."

Matrix-Switch Operation

Fig. 2 illustrates the time relation and approximate current waveforms for the inputs and outputs of the matrix switches. The Z or inhibit current applied to a memory plane and the sense output voltage are also shown.

The first three current waveforms are the inputs to the X-matrix switch during read time. Two identical sets of five bias drivers are controlled by the *thousands* and *hundreds* coders, which in turn are independently controlled by the *thousands* and *hundreds* address registers. Each binary-coded decimal digit input to a coder selects a unique combination of three of the associated five bias drivers. Each of the three drivers emits the BIAS 1 current waveform of Fig. 2(a). The other two bias drivers in each set of five are turned off when PB1 terminates and emit only PRE-BIAS 1 current as in Fig. 2(b). When PRE-BIAS 1 is turned off, only one of the 100 switch cores in the X switch remains unbiased.

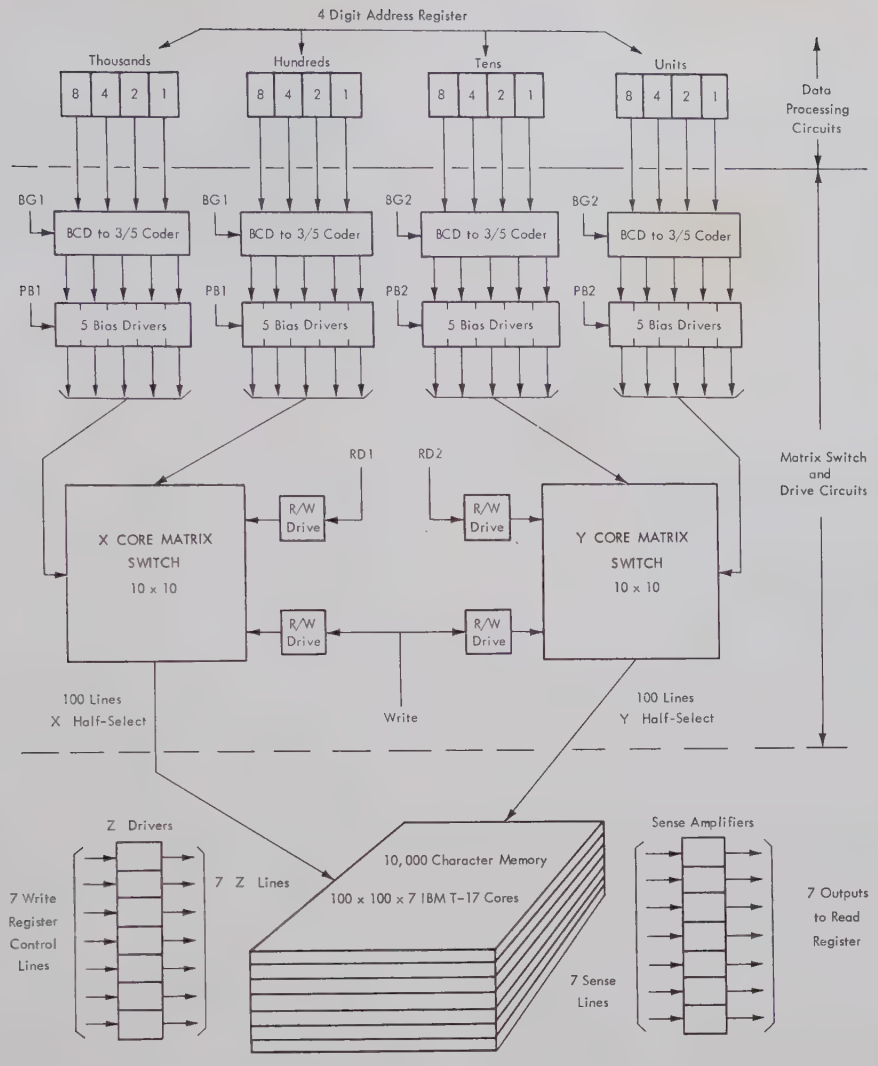


Fig. 1—Memory-system block diagram.

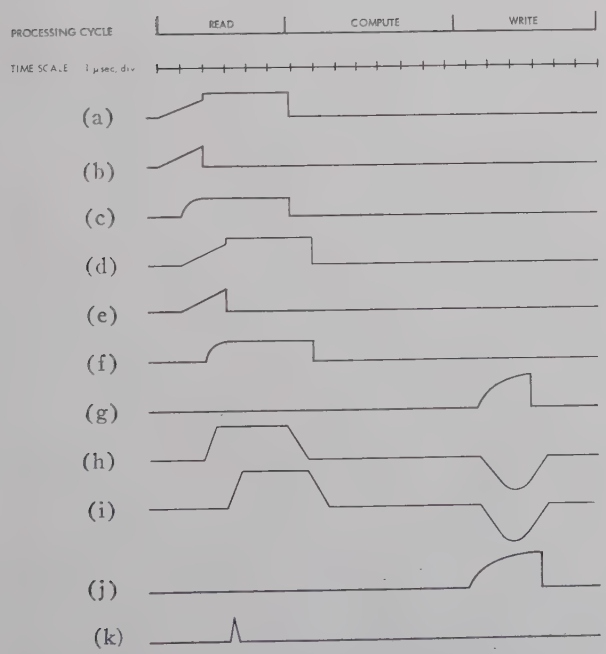


Fig. 2—Timing diagram showing processing cycle and principal memory current waveforms. (a) BIAS 1, 425 ma, (b) PRE-BIAS 1, (c) READ 1, 2 amperes, (d) BIAS 2, 425 ma, (e) PRE-BIAS 2, (f) READ 2, 2 amperes, (g) WRITE, 2 amperes, (h) *X* half-select, 250 ma, (i) *Y* half-select, 250 ma, (j) *Z*, 250 ma, (k) sense output 30 mv.

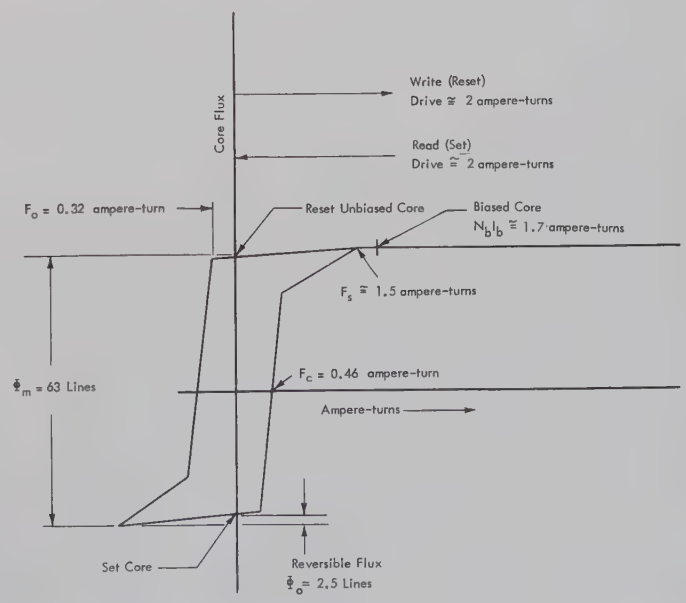


Fig. 3—Hysteresis operating curve of the IBM T-24 switch core.

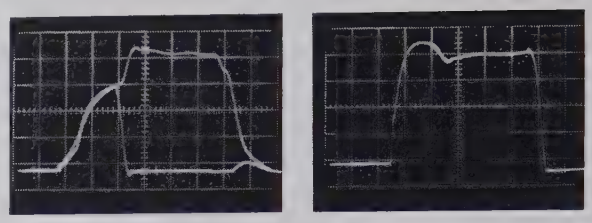
(See Fig. 3.) The unbiased core is subsequently "set" by the READ 1 current, producing a half-select read-current pulse in the memory array.

The PB1 signal in Fig. 1 is ORed with coder outputs to each *thousands* and *hundreds* bias driver. Thus, all ten bias drivers are turned on unconditionally at the start of the BIAS 1 and PRE-BIAS 1 current waveforms. Each bias-driver current biases a unique set of 40 switch cores, the sets being determined by a method described in following sections. Similarity of the bias windings insures that all drivers are equally loaded during the pre-bias period. The effective load inductances are therefore equal and limit the initial-rise rate of each bias current to approximately 200 ma/μsec. This effective inductance results from the energy required to establish the relatively small but finite increase of the magnetic field in the 100 switch cores as they are dirven from the "Reset Unbiased Core" state toward the "Biased Core" state (Fig. 3).

Each core in the matrix switch is equally driven by the mmf of a unique combination of four bias currents, two out of each set of five. Thus, during pre-bias, the total bias mmf on each core is four times the mmf applied by one bias driver. The final value of each individual bias mmf is designed to be barely sufficient to keep a switch core in the reset saturated region of the hysteresis loop when the READ 1 setting current reaches its final value. READ 1 current applies a setting mmf, magnetically opposite to that of the bias mmf, to every switch core in the matrix. After the rising bias currents reach one-fourth of their final value, READ 1 current is turned on. It does not immediately drive any cores out of the saturated reset region because its mmf is counteracted by the sum of the bias mmfs.

The READ 1 driving circuit couples equally to each of the ten bias circuits. This coupling is in a direction such that the read and bias circuits effectively aid each other in turning on current, thus producing the first sudden change in slope of the bias and pre-bias current in Fig. 4. (The BIAS 1, PRE-BIAS 1, and READ 1 current waveforms precede respectively the BIAS 2, PRE-BIAS 2, and READ 2 waveforms of Fig. 4 by 1.0 μsec; but otherwise they are nearly identical.) This coupling also causes the READ current to rise quickly to its final value, or even to a slightly larger value.

The ten bias currents are allowed to rise until they reach a point, experimentally determined, in the region of 0.6 their final value. Here the total magnetic energy of the read circuit and ten bias circuits approximates the magnetic energy which exists with six selected bias currents at their final value, four unselected bias drivers completely off, and the READ current at its final value. At that point, the four unselected bias currents are turned off by terminating the PB1 signals to all bias drivers. The coupled magnetic energy of the unselected off-going bias currents drives the selected on-going currents rapidly to their final value (Fig. 4). The READ



(a) (b)

Fig. 4—Matrix-switch input current waveforms. Horizontal, 1 μsec/div; vertical, 100 ma/div. (a) BIAS 2 and PRE-BIAS 2 currents superimposed. (b) READ 2 current through one of four parallel transistors.

current decreases slightly at this time because of its net opposite-sense of coupling with the switching bias currents.

The bias winding pattern is such that with four bias currents off (two out of each set of five) only one core in the matrix switch remains unbiased. This core is therefore driven in the set direction by the total mmf applied by the READ current. The READ mmf is counteracted in each of the 99 unselected cores with one to four units of biasing mmf. The READ current in the primary winding, by transformer action, produces an output half-select memory read current in the selected-core secondary winding. The rise time of this current pulse is approximately equal to the fall time of the off-going bias currents.

Ideally, the total magnetic energy of the four off-going bias circuits is completely absorbed by the other six bias circuits. As no change in total magnetic energy is required, instantaneous switching of the bias currents is possible. However, leakage flux, drive-transistor decay time, and a slight variation in turn-off delay among the bias drivers prevent ideal switching and limit the output current-rise time to between 0.2 and 0.3 μsec.

As a practical matter, the time at which switching occurs with the least magnetic-energy change is determined experimentally by varying the turn off of PB1 until a minimum output-current rise time occurs. Optimum memory-sense output signals result only if the rise time of the Y half-select pulse, staggered 1.0 μsec after the similar X half-select pulse, is less than 0.4 μsec. This rise time is impossible to achieve with the low driver voltage employed, if the READ pulse goes on after PRE-BIAS turns off. With such timing, the half-select rise times determined experimentally are 0.8 μsec, approximately three to four times the rise time achieved by the method of timing described.

The selected core is switched or set along the steep left-hand edge of the hysteresis loop (Fig. 3) during the output current pulse. This pulse [Fig. 2(h)] is terminated by turning off the READ current. The six selected bias currents are turned off immediately following the READ current termination.

Thus far, we have described the operation of the X-matrix switch in producing the X half-select memory-

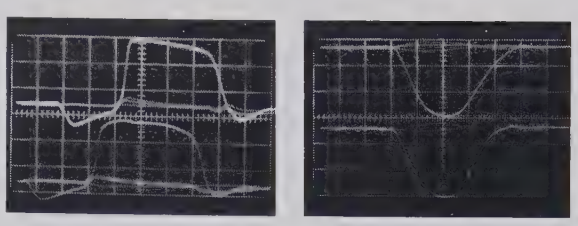
read-current pulse. In a like manner, the currents shown in Fig. 2(d)–(f) produce the Y half-select output from the Y-matrix switch [Fig. 2(i)]. The Y half-select read pulse occurs 1.0 μsec after the X half-select pulse. This 1.0-μsec delay has the effect of reducing the noise from which a memory-plane sense amplifier must detect the "one" signal.[4]

To read from the core memory, the X-Y switch output pulses must overlap each other by 1.0 μsec. Thus, a full select current is provided at the selected address for 1.0 μsec, the switching time of the memory cores. Fig. 5 shows typical current waveforms of the two half-select pulses and the output from an unselected switch core. The peak unselected output is about 17 per cent of a half-select output or 8.5 per cent of the full select output; six half-select lines from switch cores having only one unit of bias mmf, carry this maximum unselected output. Error-free operation with ±7.5 per cent margins on all drive currents with the worst case pattern in the memory indicates that this amount of noise is tolerable. The negative going current just preceding the read half-select current is caused by pre-bias current resetting core flux not completely reset at write time; this reset is desirable for stabilizing the switch core magnetization cycle. The memory-sense winding output shown in Fig. 6 is for worst case, disturbed, "ones," and "zeros."

All matrix-switch currents drop to zero during compute time with the selected switch core left in the set condition, as indicated in Fig. 3. All other cores remain in the reset condition. Inasmuch as memory-address register contents are not required for the remainder of the cycle, they may be changed during this interval.

At write time, reset pulses are driven through all 100 cores of both switch core arrays simultaneously. Coincident current pulses are emitted from the previously set switch cores which have been effectively storing the memory address. The inductance affecting the matrix write driver is about 8 μh, much greater than the inductance affecting the read driver. Such is the case because the switch cores are not saturated at write time as they are when the read driver is turned on. This results in a slowly-rising WRITE pulse, as in Fig. 2(g). Consequently, the half-select write currents out of the matrix switches have a rise time of about 1.2 μsec. (See Fig. 5(b).) The half-select write pulse must be brought up to the 250–300 ma level for 1.0 μsec in order to switch the memory cores properly. This cannot be done unless sufficient flux is set in the selected switch core at read time. The read half-select pulses must, therefore, be about three μsec long instead of the one- or two-μsec pulses that are otherwise sufficient to switch the memory cores at read time.

Because all bias drivers in each matrix switch are turned on by the corresponding PB signals, the resultant mmf will complete the resetting of a previously selected switch core. This permits turning off the WRITE currents one to two μsec earlier than would ordinarily be necessary. Thus, the time needed to set up the bias currents is not wasted, for it would otherwise be required to complete the resetting of the matrices. Furthermore, it is convenient to use the pre-bias period for setting the memory-address register.



(a) (b)

Fig. 5—Matrix-switch output current waveforms. Horizontal, 1 μsec/div; vertical, 100 ma/div. (a) Read outputs from biased and unbiased switch cores. (b) Write outputs from biased and unbiased switch cores.

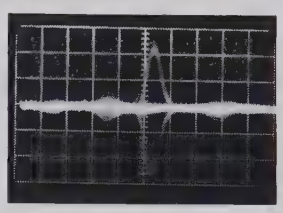


Fig. 6—Core memory-sense output from 5000 cores of 10,000 memory-core plane. Worst case zeros and ones superimposed. Horizontal, 1 μsec/div; vertical, 10 mv/div.

The BCD-3/5 Coder Design

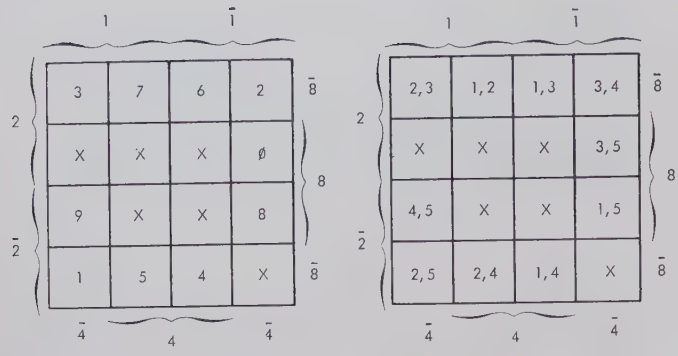
The design of the four identical coders which convert the memory address from BCD form to a 3-out-of-5 (3/5) representation is described in Fig. 7. Fig. 7(a) is a Veitch diagram or Karnaugh map of the BCD code in which $\bar{1}\ 2\ \bar{4}\ 8$ represents decimal zero.[5] Unused redundant combinations are indicated by the conventional X's. Examination shows that each bias driver must be ON for six of the ten BCD combinations. It is equivalent to say that each bias driver must be OFF for four of the ten BCD combinations.

OFF functions $\overline{BD1}$ to $\overline{BD5}$ for the five bias drivers are the simplest to derive. Since a negative signal turns the bias drivers ON, a positive output code is required for the OFF functions. The matrix switch-core windings and their interconnections require that a unique 3-out-of-5 drivers must be ON for each BCD digit and 2-out-of-5 drivers must be OFF. Therefore, it is necessary and sufficient to derive a positive-output 2-out-of-5 coder for the OFF functions.

As shown in Fig. 7(b), the coder is designed by superimposing the Veitch diagrams of five output functions onto a single map. Bias driver OFF functions $\overline{BD1}$ to $\overline{BD5}$ are designated by the digits 1 to 5, not to be

[4] M. A. Alexander, R. Rosenberg, and R. Stuart Williams, "Ferrite core memory is fast and reliable," *Electronics*, vol. 29, pp. 158–161; February, 1956.

[5] Montgomery Phister, Jr., "Logical Design of Digital Computers," John Wiley and Sons, Inc., New York, N. Y.; 1959.



(a) (b)

BCD	Bias Drivers	
	ON	OFF
Ø	124	35
1	134	25
2	125	34
3	145	23
4	235	14
5	135	24
6	245	13
7	345	12
8	234	15
9	123	45

(c)

$\overline{BD1} = \bar{1}\cdot\bar{2} + 2\cdot 4$

$\overline{BD2} = 1\cdot\bar{8}$

$\overline{BD3} = 2\cdot\bar{4} + \bar{1}\cdot 2$

$\overline{BD4} = \bar{1}\cdot\bar{4}\cdot\bar{8} + \bar{2}\cdot 4 + 1\cdot 8$

$\overline{BD5} = 8 + \bar{2}\cdot\bar{4}$

(d)

Fig. 7—Development of the Boolean functions of the BCD-to-3/5 coder. (a) Map of binary-coded decimal digit. (b) Map of 5 bias driver OFF functions superimposed. (c) Truth table for bias driver functions. (d) Boolean expression of BCD-to-3/5 coder.

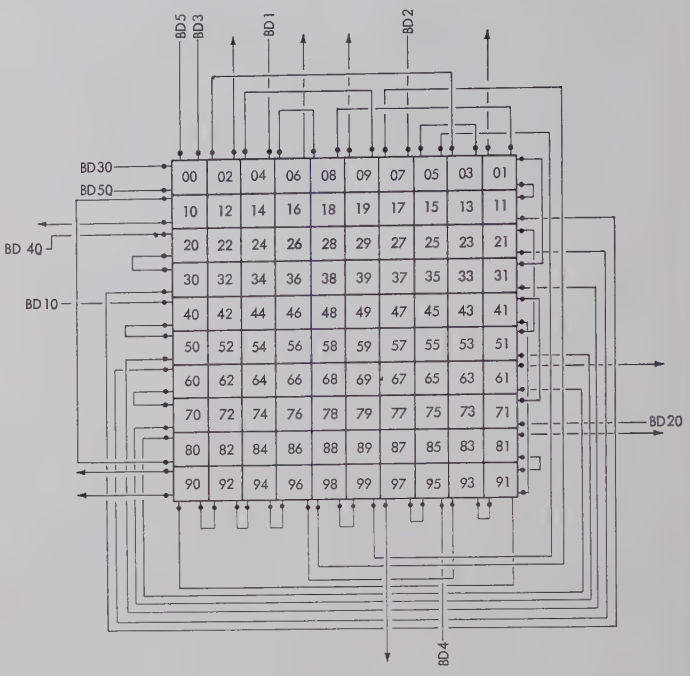


Fig. 8—Layout of a 100-core matrix switch.

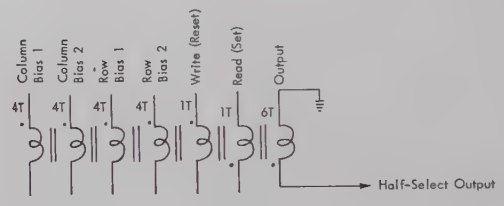


Fig. 9—Winding requirements for an individual switch core.

confused with the BCD notation or the decimal notation existing in the same figure. Mapping is a trial-and-error process. It involves arranging for the greatest symmetry and somewhat tempering the mapping to suit the logic circuits to be used. This particular coder was built with mesa-drift NOR logic blocks having a maximum of three inputs per block.[6] Each of the five digits (1–5) must superimpose once and only once on each of the other four. Only two digits can exist in each square of the Veitch diagram. The coder mapped in Fig. 7(b) was derived after several trials. It was the most economical NOR logic-block implementation. Fifteen NOR logic blocks are required for each of the four coders. Truth tables for the corresponding ON and OFF functions are given in Fig. 7(c), and the Boolean expressions describing the OFF functions to be implemented are listed in Fig. 7(d).

Address Coding in the Matrix Switch

Fig. 8 illustrates a typical matrix-switch layout and the bias-winding interconnections. Each of the blocks in the 10×10 array indicates a switch core. Fig. 9 shows the windings on a single switch core. Assuming Fig. 8 to be the Y matrix, the two digits in each block indicate the tens and units digits of the four-digit memory address. The thousands and hundreds digits of the address are similarly associated with the X matrix.

Every row and every column of switch cores is wound in series by two separate but identical bias windings. Thus, each switch core is threaded by four separate but identical bias windings. The current in any one winding is sufficient to bias the core OFF, thus inhibiting an output pulse at read time. The 20 "column" bias windings of the Y matrix are interconnected so that four windings through four separate columns are driven in series by each of the bias drivers BD1, BD2, BD3, BD4, BD5. The 20 "row" bias windings are similarly interconnected.

The two bias drivers associated with each column are indicated by Fig. 7(c) in the OFF column of the truth table. For example, the left-hand column is coded for the units-digit zero. All of the addresses with zero for the units digit are half-selected by switch cores in this column. According to the truth table, BD3 and BD5 should bias this column, and Fig. 8 shows this to be so.

One and only one column is associated with a decimal

[6] R. A. Henle and J. L. Walsh, "The application of transistors to computers," Proc. IRE, vol. 46, pp. 1240–1254; June, 1958.

address digit and the corresponding pair of OFF bias drivers. The three other bias drivers will be ON. Only one column remains unbiased by the column windings; all other columns have one or two of their column bias windings activated. Similarly, only one row of cores is unbiased by the horizontal windings. Therefore, only the core at the intersection of the unbiased row and column is unbiased. The read or set winding threads all of the 100 matrix-switch cores in series. When the read winding is energized, only the one unbiased switch core will emit a half-select current pulse. All other switch-core outputs are inhibited by 1, 2, 3 or 4 units of bias magnetizing force.

Physical Construction of the Matrix Switch

Fig. 9 indicates that each switch core requires seven windings: four bias, one read, one write, and one output. The cost of a matrix switch would be prohibitive if individual windings were necessary for each switch core. However, the winding scheme illustrated in Fig. 10(a) for the 20 pairs of row and column windings, plus the symmetry of the winding scheme, makes the matrix construction economically justifiable.

Each column and row is wound with a double wire so that a single winding operation produces the required two separate windings. The physical direction of the current is reversed through adjacent cores; this is corrected by means of the output winding connection.

The set and reset windings are identical to each other and are wound by threading a double wire through all 100 cores, alternating direction on adjacent cores as in the winding of Fig. 10(b). The opposite polarity of magnetization by the set and reset windings is obtained by driving the READ and WRITE currents through them in opposite directions. The output windings must be individually wound.

Fig. 11 illustrates the physical appearance of a single matrix switch. Fig. 12 illustrates the compact assembly of both switches in a modified memory-core frame, and Fig. 13 shows the assembly of the two matrix switches and 10,000-character core memory.

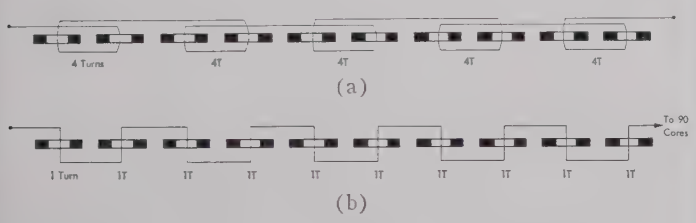


Fig. 10—Matrix winding methods. (a) Cross section of row of 10 switch cores showing bias winding pattern. (b) Cross section of row of 10 switch cores showing read (Set) or write (Reset) winding pattern.

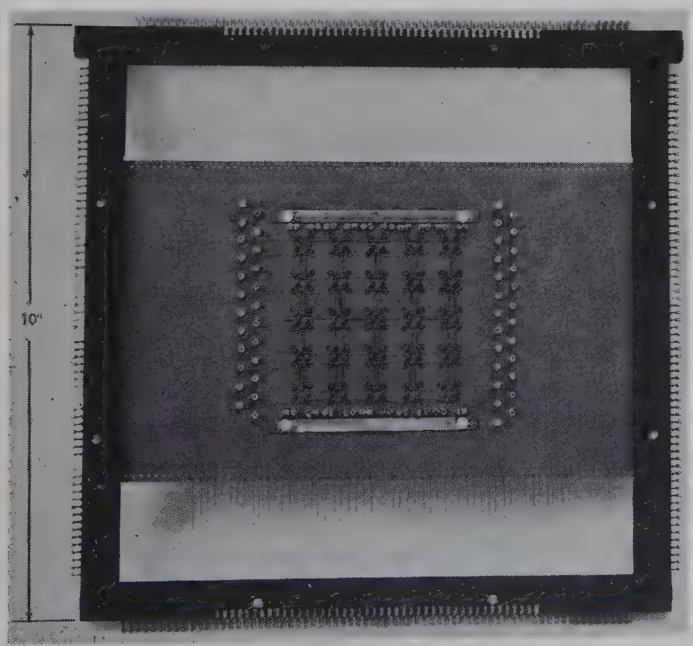


Fig. 11—Ferrite-core matrix switch with 100 output lines. (Input lines and bias interconnections are not shown.)

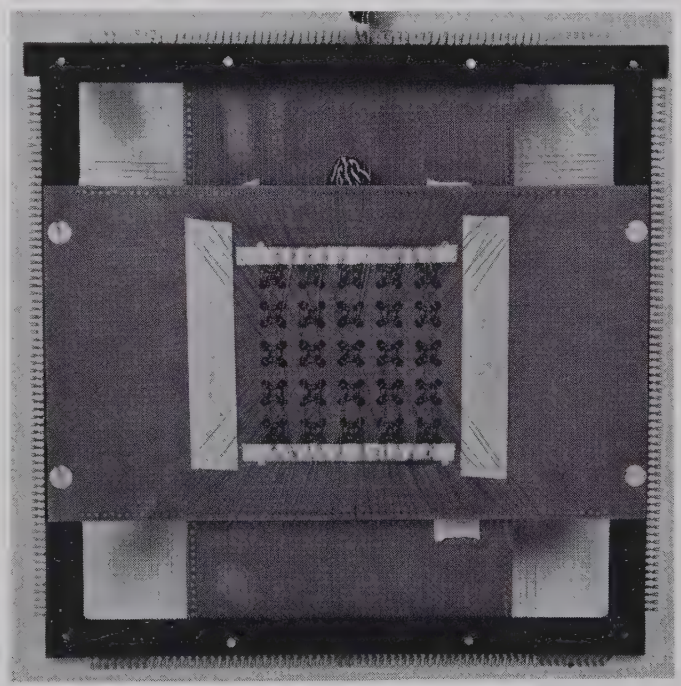


Fig. 12—Complete assembly of two ferrite-core matrix switches.

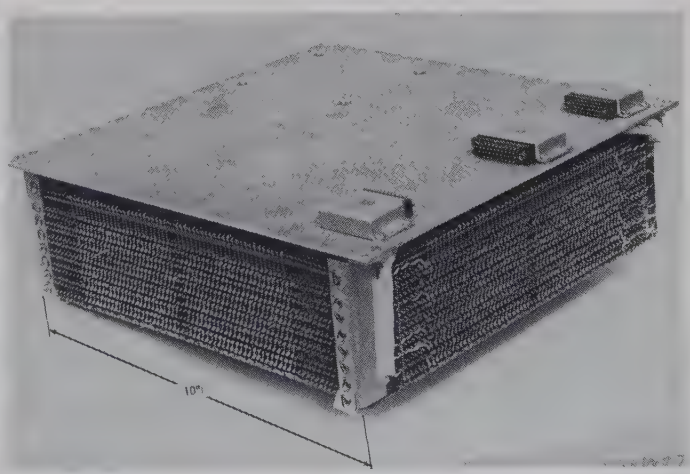


Fig. 13—10,000-character memory assembled with matrix switches.

Design of the Matrix-Switch Windings

Secondary Winding

The first step in designing the matrix switch, assuming a switch core and memory-plane assembly have been chosen, is to calculate the minimum turns required of the switch-core output winding N_2. This is essential because the lowest drive requirements will accompany the lowest N_2. It is also desirable to minimize the unselected output noise pulses which are proportional to N_2.

The data of Tables I and II adequately describe the T-24 switch core and the T-17 memory core operating at 40°C.

TABLE I

Typical Characteristics of the IBM T-42 Ferrite Switch Core at 40° C.

I.O.		0.069 inch
O.D.		0.130 inch
Height		0.066 inch
Φ_m		63 lines (total flux capacity)
Φ_0		2.5 lines (reversible flux)
H_0		0.53 Oersted
Sw		0.60 Oersted-μsec
Sw	$\triangleq$	$Ts(H-H_0)$
A		0.0260 cm^2 (cross-sectional area)
L		0.00762 m (mean circumference)
H_c	$\cong$	$H_0\left(1+\frac{\text{O.D.}-\text{I.D.}}{2\text{ I.D.}}\right)$ 0.76 Oersted
F_c	$\cong$	0.46 Ampere-turn (coercive magnetizing force)
F_0	$\cong$	0.32 Ampere-turn (threshold magnetizing force)

TABLE II

Typical Characteristics of the IBM T-17 Ferrite Memory Core at 40° C.

I.D.	0.030 inch
O.D.	0.050 inch
Height	0.012 inch
t_s	0.8 to 1.0 μsec
$Ipr=Ipw$	250 ma
$Ir=Iw$	500 ma
Φ_m	5.2 lines (total flux capacity)
Φ_0	0.08 lines (reversible flux for half select magnetizing force)

The equivalent circuit of a switch core driving a single memory half-select line is given in Fig. 14. The load consists of the following four parts with flux units in lines or maxwells:

Φ_1 = Reversible flux of 693 half-selected memory cores

Φ_2 = Total flux capacity of seven selected memory cores (This may vary between one and seven, depending on the character stored at the memory address)

$\Phi_3=10^{+8}L_3I_{pr}$ Maximum reversible flux stored in the half-select-line air inductance

$\Phi_4=10^{+8}\int V_r dt=10^{+8}\int I_{pw}R_4 dt$ Flux needed to maintain the required voltage drop across the half-select-line resistance.

No resistor load other than the line resistance is required. This saves the cost of installing two hundred load resistors and also reduces the drive requirements. A flatter pulse top is also achieved since the flat-top "droop" time constant is L_T/R_T where L_T and R_T are total equivalent quantities for the secondary circuit including the mutual inductance of the secondary winding. The load flux requirement is

$\Phi_1+\Phi_2+\Phi_3+\Phi_4=\Phi_L=354$ lines (total flux load)
Select $N_2=6$ turns ($N_2\Phi_m$ must be greater than Φ_L)
$N_2\Phi_m=6$ turns $\times 63$ lines/turn $=378$ lines (output flux capacity).

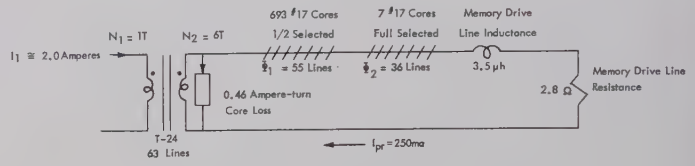


Fig. 14—Equivalent circuit of selected switch core.

Primary Winding

The primary matrix READ (set) and WRITE (reset) current requirement is[7]

$$I_1 = (N_2I_2a + F_c)\frac{1}{N_1}. \tag{1}$$

The switching constant S for an unloaded core is defined here by

$$\frac{dB}{dt} = S(H - H_c) \cong \frac{2Bm}{T_s} \quad \text{for} \quad H \gg H_c$$

$$S \cong \frac{2Bm}{Ts(H - H_c)}.$$

The effect of the switching constant on the loaded switch core operation is represented by the factor a in (1):

$a=1+R/N_2^2\cdot L/AS$
R = Secondary load resistance
L = Mean circumference of core
A = Cross sectional area of core

To evaluate AS/L, let $H-H_0$ approximate $H-H_c$. Then,

$$\frac{AS}{L} \cong \frac{A \times 2Bm}{LT_s(H - H_0)} \cong \frac{\Phi_m}{LS_w} \qquad (S_w \text{ defined in Table I})$$

For the circuit designed, $a=1.046$. We set $N_1=1$, the minimum, in order to maximize the speed of the memory. The penalty paid for this speed is an increased cur-

[7] D. A. H. Brown, "Approximate methods for calculating the behavior of square loop magnetic cores in circuits," *Electronic Engng.*, vol. 31, pp. 408–411; July, 1959.

rent drain from the power supply. Then from (1), $I_1 = 2.03$ amperes.

Bias Winding

In order to inhibit core switching, the bias magnetizing force must counteract the READ current mmf, $N_1 I_1$.

$$\therefore \quad N_b I_b \geqq N_1 I_1 - F_0$$

$$N_b I_b \geqq 2.03 - 0.32 = 1.71 \text{ ampere-turn (at 40°C)}$$

We have selected $N_b = 4$.

$$\therefore \quad L_b = 0.425 \text{ ampere (at 40°C)}.$$

Transient Behavior of Bias Currents

All bias drivers for each matrix are turned on simultaneously by the pre-bias signals PB1 and PB2 in Fig. 1. The symmetry of the bias windings insures that all bias currents will rise at the same rate. All switch cores are initially at the reset position on the hysteresis curve and are driven towards the biased condition as indicated in Fig. 3. The rise rate of the bias current is calculated as follows:

$\frac{di_b}{dt}$ = Initial rate of rise of bias current

$V_c = 10$ volts total drive voltage

$N_b = 4$ turns per core per bias winding

$m = 4$ bias windings per core

n = Total of 40 cores per bias winding

$\Phi_0 = 2.5$ lines reversible flux (Fig. 3, Table I)

F_s = Approximately 1.5 ampere-turns (Fig. 3)

$\sum N_b i$ = Total bias magnetization per core

$$\frac{V_c}{N_b n} = 10^{-8} \frac{d\Phi_0}{dt} \text{ Volts per turn of bias winding at } t_0 \quad (2)$$

$$\frac{d\Phi_0}{dt} = \frac{d\Phi_0}{d(\sum N_b i)} \frac{d(\sum N_b i)}{dt} . \quad (3)$$

Since all bias currents are equal,

$$\sum N_b i = m N_b i_b \quad (4)$$

$$\frac{d(\sum N_b i)}{dt} = m N_b \frac{di_b}{dt} \quad (5)$$

$$\frac{d\Phi_0}{d(\sum N_b i)} \cong \frac{\Phi_0}{F_s} . \quad \text{(Fig. 3)} \quad (6)$$

Combining (3), (5) and (6),

$$\frac{d\Phi_0}{dt} \cong \frac{\Phi_0}{F_s} m N_b \frac{di_b}{dt} . \quad (7)$$

Combining (7) and (2),

$$\frac{di_b}{dt} \cong \frac{10^8 V_c}{n m N_b^2} \frac{F_s}{\Phi_0} . \quad (8)$$

Substituting into (8) from the previously listed values,

$$\frac{di_b}{dt} \cong 234 \text{ ma/}\mu\text{sec}.$$

By direct measurements of Fig. 4(a),

$$\frac{di_b}{dt} \cong 150\text{–}200 \text{ ma/}\mu\text{sec}.$$

The discrepancy is easily attributed to the inaccuracy of the straight line approximation to the hysteresis loop saturation region (Fig. 3) as expressed in (6).

Transistor Driver Circuits

All currents required to drive the matrix switch and memory are derived from a main power supply of 10 to 12 volts, plus a 3-volt cascaded bias supply. Bias drivers are equipped with a single alloy *p-n-p* output transistor as in Fig. 15. The *Z* or inhibit drivers are similar. Set and Reset drivers are equipped with four such output transistors in parallel. The collector clamp is required to dissipate the stored magnetic field in the inductive load when the transistor is turned off.

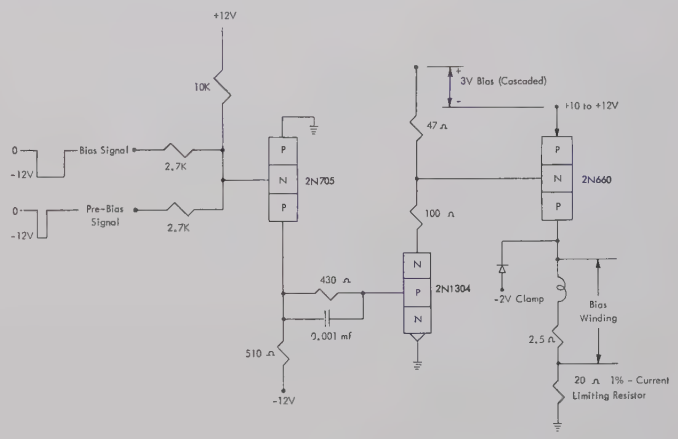


Fig. 15—Bias driver circuit.

Even at such a low voltage, the Set and Reset drivers act as constant current sources. The series load resistance reflected back to the driver of the primary in the equivalent circuit of Fig. 14 is only 0.078 ohm, whereas the total limiting resistance is about 5 ohms.

The low-voltage power supply was selected so that all drive currents, including the *Z* currents, could be economically derived. Comparatively low-cost alloy transistors are used in all drivers, and the power-supply voltage is limited by the transistor specifications. Besides being economical, one power supply is desirable because all drivers can then be readily compensated for temperature variations in the memory core requirements. Also, with a common power source *X*, *Y* and *Z* currents vary together, permitting greater power-supply voltage tolerances. The supply voltage is controlled to vary from about 10 volts at 40°C to 12 volts at 10°C. Maximum power dissipation of the complete matrix-switch system, plus the *Z* drivers, is 60 watts-12 volts, 5 amperes at 10°C.

Packaging

Fig. 16 illustrates the packaging of the matrix-switch driving circuits together with the 10,000-character memory, the sense amplifiers, timing circuits, and miscellaneous control circuits. All circuits associated with the memory (Fig. 1), except the power supply and the memory-address register, are on 57 pluggable printed circuit cards on this $14'' \times 18'' \times 5\frac{1}{2}''$ chassis.

The principal components required are:

Diodes	214
Transistors	209
Resistors	748
Capacitors	71
Filter Capacitors	33
Pulse Transformers	14

Results and Conclusions

A low-cost ferrite-core matrix switch and drive system has been developed for a 10,000-character, 7-bits-per-character memory which operates in a 20-μsec machine cycle over an ambient temperature range of 10°C to 40°C.

The reliability of the matrix-switch drive system is comparable to that of more expensive drive systems operating with the same memory planes. This is demonstrated by testing the memory-sense output at every address over the 10°C to 40°C range with the worst case pattern in each array. During the test, separate variable power supplies are provided for the inhibit drivers (V_Z) and for the R/W and bias drivers (V_{xy}). Under these conditions, the system can tolerate ± 7.5 per cent variations in Vz and V_{xy} without making any type of error.

A novel coding scheme and matrix winding technique have resulted in a very simple matrix construction. The coding scheme also permits a saving in the number of drivers so that a total of only 24 drivers is required for two 100-output matrix switches.

The setting READ current is turned on when all bias drivers are on (during pre-bias), and the leading edge of the output read pulse to memory is formed by turning off the appropriate bias drivers. Because bias drivers can be turned off very rapidly, this results in an exceptionally fast rise time (0.2 to 0.3 μsec) for a memory supply voltage of only 10 to 12 volts.

Pre-biasing also completes the reset of the switch core used in the previous cycle. This permits shortening of the resetting WRITE current pulse. In addition, the memory-address access time can be overlapped with the pre-bias time, because the switch drives during pre-bias are independent of the memory address. Use of a current-pulse bias, rather than the more common dc bias permits the memory-address register to be counted or otherwise employed during the compute and write portions of the memory cycle. Also, the standby power of the memory is negligible. Alloy junction transistors perform all matrix drive-current switching with maximum dissipation in any transistor being 100 mw at 40°C.

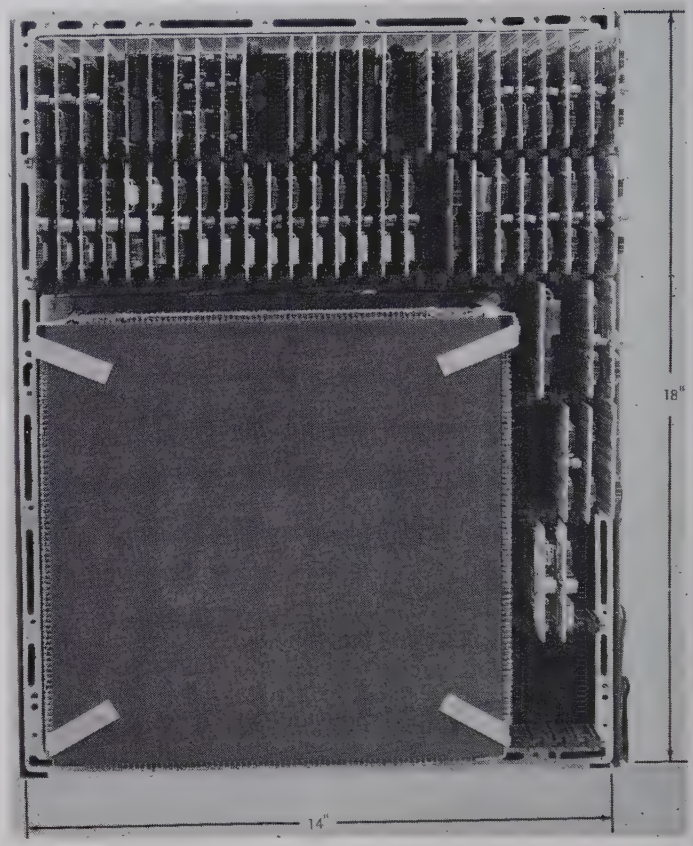


Fig. 16—Complete 10,000-character memory and associated circuits.

Acknowledgment

The author is pleased to acknowledge the contributions of L. R. Harper and W. E. Burns, who suggested operating the matrix with 3/5 coding. Harper also provided valuable assistance in the design of circuits for a preliminary model of the memory. A. C. Slutman assisted in the logical design, layout and testing, and J. D. Bartlett aided in the construction of the memory and test circuitry.

Serial Matrix Storage Systems*

M. LEHMAN†, MEMBER, IRE

Summary—Coincident-current techniques, usually associated with parallel ferrite-core stores, may also be used for the operation of serio-parallel or purely serial memories. After outlining, in block diagram form, one possible physical realization of a serial system, the paper examines the conditions under which such a store is economically justified. The distinguishing feature of the system discussed is that coincidence is established in the memory matrix between two currents representing an address signal and a time signal, respectively. Studies of the characteristics and economics of serio-parallel devices are however not reported in detail.

It is shown how the properties of the time-controlled serial store may lead to the adoption of a word-asynchronous design for serial digital computers. In such a machine, timing is not controlled or determined by limited store access. As examples, the paper indicates how the serial techniques facilitate the incorporation into small serial computers, of autonomous transfers, automatic floating point operations, high speed multiplication, division and shift orders and asynchronous transfers between, say, a high speed store and a magnetic drum.

I. Introduction

COINCIDENT-current storage systems are normally regarded as being essentially parallel devices and have consequently found their most natural application in parallel digital computers. Where ferrite-core stores have been used in serial machines, advantage has been taken of the parallel availability of bits, to match the maximum reading rate obtainable for a given expenditure to the speed of the remainder of the machine. The latter may thus operate economically with a bit-repetition rate considerably in excess of that which could be used were the store to be operated serially. The one megacycle serial Mercury computer, for example,[1] uses a ten-microsecond, coincident-current core store, reading (or writing) a ten-bit "Short Word" to (from) a tapped, electromagnetic delay line. The individual bits emerge from (are fed to) this line serially at one-microsecond intervals for transmission to (from) the arithmetic unit or other destinations (sources).

The size of the bit group processed in parallel in an n-bit storage scheme may vary from "one" for the purely serial device, to "n" for the wholly parallel scheme. The optimum size will normally be determined by the need to strike a balance between economic considerations and the relative speeds of the machine and of possible memory elements and circuits. Logical design considerations may however also exert an influence on the final selection of the group unit to be used. Groups of from four to eight bits have, for example, been used in a number of machines which operated on alphanumeric or decimal characters.

Having selected the group size, a further alternative presents itself. In a serial system with destructive readout, it would appear natural to circulate and rewrite each individual bit as it is read. It is however possible to store the bits in, say, a ferrite-core shifting register from which the word as a whole may be rewritten in parallel on completion of the reading process. In this way the maximum clock frequency compatible with a given store may be almost doubled.

The purpose of the present paper is to show how in certain applications it may prove advantageous to use serial, time-selection techniques in operating a matrix store. The feasibility and logical and circuit design details of such a memory system will ultimately depend on the bit rates and the store cycle times which are economically justifiable and which it appears desirable to achieve. The study does *not* investigate the systems design criteria which justify or demand the use of a purely serial store. It is proposed merely to outline some of the advantages which may accrue when such a scheme is used.

II. Possible Systems of Time-Selection

Any parallel storage system may be converted to pseudo-serial operation by gating the outputs from the store with appropriate clock pulses. Similarly the high-level output of an inhibit driver could be "steered," sequentially, to the various planes for writing. Such a scheme could save some equipment by reducing the number of read amplifiers and inhibit drives, though the problems arising from gating at signal level might well neutralize any saving. The net gain in logical flexibility and the total reduction in cost would however at best be relatively small, since such a system is still essentially parallel.

The characteristic feature of the serial store described in the present paper arises from the fact that coincidence is established *in the store matrix*, between the whole address signal and between a time signal. The scheme is illustrated in outline in Fig. 1, which may be contrasted with the more usual parallel scheme as applied to serial computers and which is illustrated in Fig. 2.

* Received by the PGEC, July 25, 1960. The study reported in this paper was carried out under the auspices of the Scientific Department, Israel Ministry of Defence.

† Scientific Department, Israel Ministry of Defence, Tel Aviv, Israel.

[1] K. Lonsdale and E. T. Warburton, "Mercury, a high-speed digital computer," *Proc. IEE*, suppl. no. 2 (Digital Computer Techniques), vol. 103, pt. B, pp. 174–183; April, 1956.

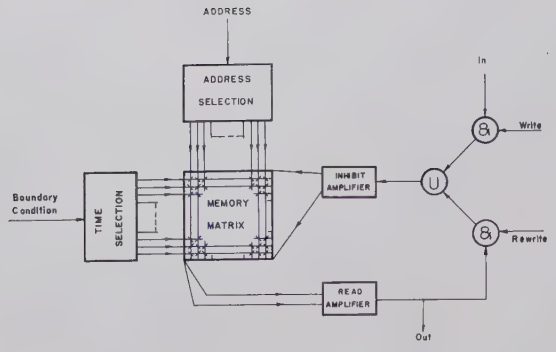


Fig. 1—The serial coincident-current store.

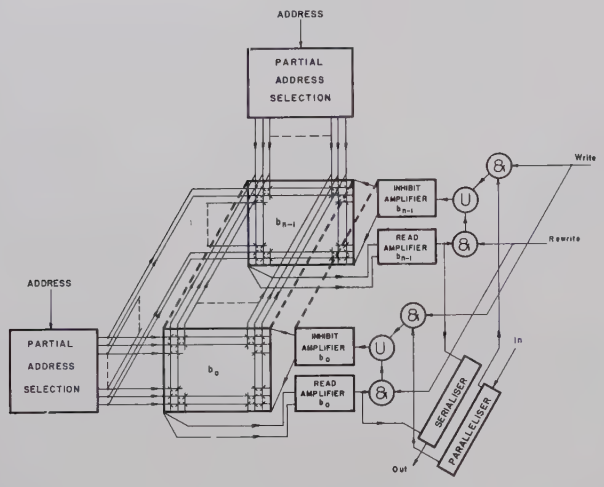


Fig. 2—The parallel coincident-current store for application in a serial computer.

The practical details of either system will depend on the type of selection system proposed.[2,3] To simplify the presentation and to avoid the confusion caused when alternatives are compared at each stage of the discussion, it is proposed to write here exclusively in terms of switching-matrix techniques. Since differences between systems tend to be marginal, a proposal to use the serial store in any particular system will in any case require further exhaustive study of the various possible selection circuits within the framework of a general design study.

In the serial, n-bit coincident-current store, the two switching matrices X and Y of the normal parallel M.I.T. system are coalesced into a "single," larger address selection matrix controlling, say, the Y wires of the main memory frame. Serial time selection of successive bits, that is of successive X wires, could then be obtained through the direct use of the machine clock pulses, from an n-place stepping switch, from an n-place counter and decoder or finally through the use of an n-core time-selection switching matrix which is in turn controlled by somewhat smaller stepping switches or counters.

The direct use of clock pulses in a word-synchronous machine is logically inflexible and is not further considered here. It might however find application in a simple computer in which cost, space, or weight factors are of more importance than speed or programming simplicity. The method appears, however, to offer possibilities in a fully word-asynchronous design. In such a machine each core access would initiate a new "word-time" or "p sequence" of duration, a function of the word length to be processed. It is hoped to study such systems in the future.

Selection of one of the other time-selection systems will depend largely on factors external to the present discussion, being chiefly connected with the circuit elements and general structure of the main equipment. Thus, a general study not related to a specific system is unlikely to yield definite results. The present study is based on the use of a second, time-selection, switching matrix.

III. The Serial Store

The system introduced with Fig. 1 may now be further particularized and Fig. 3, which is largely self-explanatory, presents one possible configuration for a serial store having a capacity of $K = abcd$ words, each of $m = pq$ bits.

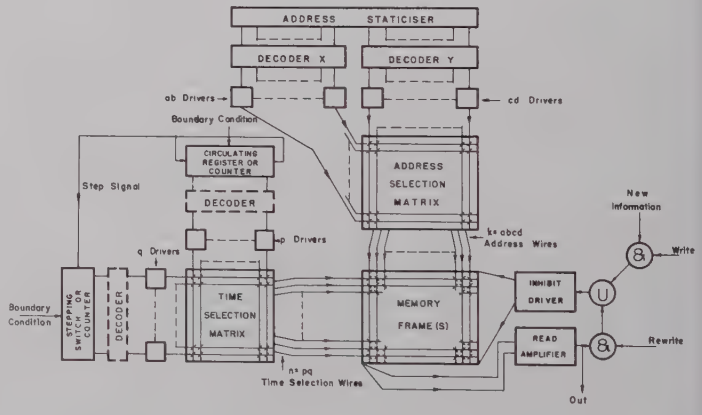


Fig. 3—The full-serial coincident-current store system.

The time-selection matrix in this circuit has been shown in rectangular form. It would be possible to use a triangular matrix as proposed by Renwick,[4] fed from a relatively simple "two-out-of-pq" counter. Design of the resultant constant-current drivers controlled by such a

[2] J. A. Rajchman, "A myriabit magnetic-core matrix memory," Proc. IRE, vol. 41, pp. 1407–1421; October, 1953.

[3] H. J. Heijn and W. C. de Troye, "A fast method of reading magnetic-core memories," *Philips Tech. Rev.*, vol. 20, pp. 193–207; February 10, 1959.

[4] M. V. Wilkes, W. Renwick and D. J. Wheeler, "The design of the control unit of an electronic digital computer," *Proc. IEE*, vol. 105, pt. B, pp. 121–128; March, 1958.

counter poses special problems however, so that in practice the scheme is not significantly more economical than the one illustrated.

The counter controlling the time-selection matrix need not, of course, be set to zero when commencing a read-write cycle, nor need the cycle start at any standard time. Thus, by the appropriate setting of the "Initial" and "End" conditions in the counter, any size group of bits from one to "n" may be made available serially, as and when required. It is precisely this asynchronous-partial-read property of the serial store which makes it so useful in the design of serial computers permitting a design flexibility not normally associated with serial machines. Examples of how this facility may be applied will be given in later sections.

The remainder of the serial store is similar in principle to the corresponding parallel system, except that basically only one each of read amplifier, circulation gate, and inhibit driver are required. In practice, however, some duplication may prove necessary and this point is further discussed in the next section.

IV. The Relative Costs of Serial and Parallel Stores

The description of the serial coincident-current store has been necessarily brief. Any comparison between alternative systems can be complete only if *fully designed memories, integrated into a complete computer system*, are considered. An estimate of the relative costs of the alternative systems may however be obtained from a more superficial comparison. It is proposed here to base the comparison on the total number of amplifiers and drivers required by either system. Such a basis is not entirely realistic since the various amplifiers required by and in each system differ in their characteristics and, hence, their cost. An amplifier count will however provide a first indication of relative costs, indicating which particular ensembles should be further investigated for the purposes of an actual design study. The differing requirements of each system for decoders, switching cores, supplementary amplifiers, gates, serializers and parallelizers must also be disregarded in a first estimate of relative costs. All these items materially affect the final cost of the memory. Their effect on relative costs is small however, and it does not appear practical to present an analysis for the general case.

For a store of capacity $k=abcd$ words each of $n=pq$ bits, a parallel coincident-current switching matrix scheme requires a minimum of $a+b+c+d$ core drivers feeding a total of $ab+cd$ switching cores. The two fully selected cores provide drives for $pqab$ and $pqcd$ data cores, respectively, of which, in each case, pq are fully selected and the remainder half selected. In addition to these selection circuits, the store requires pq read amplifiers and a similar number of inhibit drivers. Thus, *within the limitations previously set and excluding possible duplication of equipment*, the cost of the parallel store may be represented by

$$P = a + b + c + d + 2pq \text{ units.} \tag{1}$$

A similar measure for the cost of the serial system outlined in Fig. 3 is then

$$S = ab + cd + p + q + 2 \text{ units.} \tag{2}$$

The relative values of P and S for various store capacities and for word lengths of thirty-six and forty-eight bits are plotted in Fig. 4 (next page).

The graphs clearly indicate the sensitivity of the cost of the parallel store to word length. The relative cost of the serial system, on the other hand, rises much more rapidly for increases in store capacity. The precise cross-over point, however, at which economic consideration suggests a change of system, will be determined by the actual cost of the different amplifiers required, as well as by various factors not considered so far.

Of the latter, by far the most important is due to the presence in each case of differing numbers of half-excited cores. These determine the maximum back voltage which can appear on any of the driven wires (X, Y, Inhibit) and the maximum noise level on the read wire. Hence, they determine the specification of the various amplifiers and, in particular, whether certain of these require duplication. A need for duplication would of course modify the conclusion to be drawn from the curves of Fig. 4. General expressions for the number of cores threaded by each wire and for the number of these that are half selected, as a function of the store capacity and word length, are presented in Table I.

It should be noted that the read wire is normally arranged to thread the cores so that signals of both polarities appear. A degree of noise cancellation therefore occurs, and the effective maximum number of half-excited core outputs, appearing on the read wire, is less than half that indicated in the table.

What is of interest to the present study is, of course, the degree of duplication of equipment required for any serial system. This can be determined from data of the type presented in Fig. 4 and Table I correlated with information on the circuits, and more particularly, on the ferrites used. It does appear, however, that the larger time-controlled stores will require more duplication than the same size parallel store. Thus, the limit to the economic exploitation of the former technique will lie somewhat lower than the curves of Fig. 4 indicate, that is, in the region of five hundred to one thousand locations. However, for stores having a capacity less than this break-even point, the serial time-selected store offers a definite saving in equipment, in machines of appropriate speed, when compared with the fully parallel store.

Serio-parallel systems have not been included in the

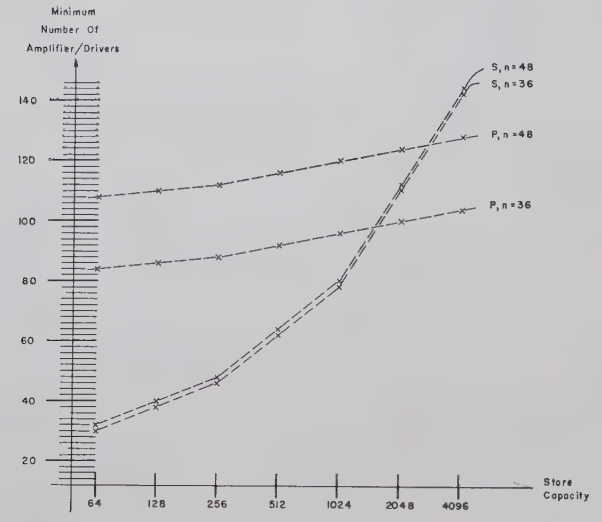


Fig. 4—The comparative requirements for drivers and amplifiers of parallel (P) and serial (S) stores for various store capacities and word lengths.

TABLE I

NUMBERS OF CORES AND HALF-EXCITED CORES ON THE VARIOUS MATRIX WIRES

Wire	Parallel Store		Serial Store	
	Cores Threaded	Half Selected	Cores Threaded	Half Selected
X	$abpq$	$(ab-1)pq$	$abcd$	$abcd-1$
Y	$cdpq$	$(cd-1)pq$	pq	$pq-1$
Inhibit	$abcd$	$abcd-(ab+cd-2)-1$	$abcd\ pq$	$abcd\ pq-(abcd+pq-2)-1$
Read	$abcd$	$ab+cd-2$	$abcd\ pq$	$abcd+pq-2$

present study, since the paper is mainly concerned with discussing and illustrating the principle of time selection. Where such combinations have been used in the past (*e.g.*, Merwin[5]) the systems have not been based on the time selection scheme. A simple count shows that for small parallel groups the differences in costs between serial and serio-parallel time-selected systems are marginal, though in a general fashion the optimum size group appears to be of the order of three or four bits. In practice, selection of a group size will be governed entirely by the store size, word length, bit repetition rate, store cycle time and problems of systems design convenience.

The remaining sections of the paper will outline some of the uses that can be made of the properties of the serial, coincident-current, time-controlled store in machine design.

V. Systems Design Aspects

High-speed stores of from thirty-two to some hundreds of locations find important applications in the class of smaller and cheaper digital computers. Machines of this type normally use a magnetic drum (or drums) for their main store. By providing a small high-speed store for the storage of current orders and data, the need for the "optimum programming" of such machines is largely removed. This simplifies their use but much time has still to be spent in the interchange of information between the high-speed and backing stores. The effect of this transfer time may be largely eliminated by the provision of autonomous (concurrent) transfer facilities.

Such a facility can be achieved in a number of ways. In its cheapest realization, the order for the transfer of information between a full track on the drum and a section of the high-speed store is combined with an order (*e.g.*, multiplication) of approximately the same duration and not requiring access to the high-speed store during its execution.[6] A more effective approach is one in which the relatively cheap serial store finds important application. Basically it is necessary to divide the high-speed store into at least two smaller sections. At any one time, one of these is designated the Program store and is used as the working store providing storage space for current instructions and data. At the same time, the second part—the Transfer store—is used as a buffer providing facilities for the autonomous transfer of information between the high-speed store and the magnetic drum, or for that matter, the peripheral equipment. A simple "Change Store System and Transfer Control" order provides for an interchange of function between the two sections of the store. Thus, information in what was previously the Transfer store may be processed while partial or final results in the other store are simultaneously transferred to the backing store or to the output channel, making room for new information or for subroutines required by the main program. The system requires that a third section of the high-speed store be fixed in the Program system to provide continuity after a "Change System" order.

To be feasible in the type of machine in which it is likely to be important, such a scheme requires the availability of a relatively inexpensive high-speed store. The present investigations have indicated, and a more detailed analysis has shown, that for a capacity of from one to five hundred locations, the serial or serio-parallel, coincident-current, time-controlled ferrite store is more economical than its parallel equivalent or the use of

[5] R. E. Merwin, "The IBM 705 EPDM memory system," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-5, pp. 219–223; December, 1956.

[6] M. Lehman, "The specification development of a cost limited digital computer," *Proc. Internatl. Conf. on Information Processing*, Paris, France, pp. 365–374, June, 1959, Butterworths, Ltd., London, Eng.; April, 1960.

either ultrasonic delay lines[7] or high-speed tracks on the magnetic drum itself.[8]

The present description of the autonomous high-speed store in serial computers has been necessarily brief, but the author hopes shortly to describe a new machine in which a serial memory and many of the other features described briefly below have been incorporated.

VI. Instruction Extraction

In one-address and two-address machines it is usually possible and desirable to store two orders in each storage location. The effective size of the high-speed store is thus greatly increased, but in most machines the facility results in a requirement for extra hardware, including a special register for the temporary storage of the second of each pair of instructions. Moreover, since the order pairs are extracted from the store together, certain programming restrictions arise. The first order may not, for example, in any way modify the second. Use of the serial or serio-parallel core store may remove both the need for extra equipment and all programming restrictions since adjustment of the initial conditions as required enables the reading out of each instruction as a separate entity. The reading of this half word together with the preceding setting up of its address will take only a little more than half a word time. By utilizing the time remaining after the reading of the order, many orders (particularly organizational, transfer control, and "short-word" orders) may then be completed in only one word time. Alternatively the remaining time may, where operation is word-synchronous, be used to read the high-speed store a second tine if, for example, a short word such as a shift number in a shift order, has to be extracted in preparation for the execution of the main order.

VII. Automatic Floating Point

It is unusual to find a serial machine permitting both fixed-point and automatic floating-point operation. Even where, as in parallel machines, such orders exist side by side, there is a considerable discrepancy between the duration of the two versions of addition and subtraction. This difference arises partly from the need for performing both exponent and mantissa arithmetic but more particularly from the need for a *relative* shift between the two operands in the event that the exponents are not equal. In most present machines, this entails an actual backward shift of the operand with the smaller exponent. Full utilization of the properties of the serial store permits a great increase in the efficiency of floating-point orders, which need be no slower than the fixed point versions.

Where both operands are stored in time-access storage systems, the exponents may be extracted and manipulated immediately after extraction and interpretation of the order. Thus, by the end of one word time, full information is available about the order that is to be performed, about the exponent of the result and about any relative shifts that may be required. The last may then be simply obtained without alteration of the operands themselves and by appropriate setting of the initial conditions in the respective time-selection control circuits. Thus, the operand with the larger exponent is read from the store in the normal way. The second operand will (unless the exponents are equal) be only partially read, the number of its least significant bits not being read being equal to the number of backward shifts that would normally have to be applied.

The process as outlined will not necessarily leave the result in standard form. Standardization would require an additional shifting phase. For normal serial shifting methods the latter would require a variable number of word times. However, where a serial store or a shifting register is used, the shift time may be reduced to one word time using the technique outlined below. The requirement for shift may also be largely avoided if "Significant Point-Arithmetic"[9] is adopted.

A difficulty that may arise in connection with the present proposal concerns the ultimate destination of the result. If the method of serial rather than parallel rewrite is to be used, the final sum or difference must either replace that operand having the larger exponent, or it must be left in a special accumulator or register for ultimate transfer to its final destination. The selection of the appropriate step to take involves questions of over-all systems design. Since this paper is mainly concerned with a discussion of the storage system, these additional details will not be pursued further.

VIII. Multiplication, Division and Shift

In the previous section it was shown how the application of differing initial conditions to two separate storage systems could be used to provide a relative shift between two numbers. This technique may be employed in a machine having, say, a high-speed store and register(s) of the serial coincident-current type to speed up multiplication and division operations by ensuring that the duration of these is only one or two word times more than the number of additions or subtractions that actually have to be performed during any particular operation.

[7] W. S. Elliott, C. E. Owen, C. H. Devonald and B. G. Maudsley, "The design philosophy of Pegasus, a quantity production computer," *Proc. IEE*, suppl. no. 2 (Digital Computer Techniques), vol. 103, pt. B, pp. 188–196; April, 1956.

[8] R. J. Proggatt, "Logical design of a computer for business use," presented at the convention on Electronics in Automation, Cambridge, Eng., June, 1957. Also, *J. Brit. IRE*, vol. 17, pp. 681–696; December, 1957.

[9] R. L. Ashenhurst and N. Metropolis, "Unnormalized floating point arithmetic," *J. Assoc. Computing Mach.*, vol. 6, pp. 415–428; March, 1959. Also, "Significant digit computer arithmetic," IRE Trans. on Electronic Computers, vol. EC-7, pp. 265–267; December, 1958.

In the case of multiplication, for example, it is possible to examine the multiplier bits, one at a time. Whenever an addition is indicated, further reading of the multiplier register is temporarily inhibited. The accumulator and the multiplicand source are then read serially with a relative shift between them of one more than the number of zeros received from the multiplier since the last addition. In this fashion the average duration of multiplication may be almost halved. If circuits are provided for selecting as multiplier that operand with fewer "ones," an even greater average speed up may be achieved.

Using similar techniques it now also proves advantageous to apply, in serial computers, the modified short-cut multiplication[10–15] and division[12–14,16] techniques recently described by several authors for application in parallel machines.

Fast shift orders may be similarly achieved. Thus, in transferring a number from one location to another any required shift may be obtained by the application of the appropriate initial conditions to the circuits controlling the time selection mechanism of one of the locations. The precise details of the circuit will depend on whether the shift is to be logical or arithmetic. The duration of the order depends on whether the number to be shifted may be transferred to a new location or whether its present location is to be unchanged. In the former case, only about one word time is required, in the latter some two, since the shift occurs during transfer to, say, a buffer register.

[10] G. Reitwiesner, "Summary Discussion on Performing Binary Multiplication with the Fewest Possible Additions," Ballistics Res. Labs., Aberdeen Proving Grounds, Note 1113; February, 1957.
[11] M. Lehman, "Parallel Arithmetic Units and Their Control," Ph.D. Dissertation, London University, Eng.; February, 1957.
[12] "On the Design of a Very High-Speed Computer," Digital Computer Lab., University of Illinois, Urbana, Rept. No. 80; October, 1957.
[13] K. D. Tocher, "Techniques of multiplication and division for automatic binary computers," *J. Mech. Appl. Math.*, pp. 364–384; August, 1958.
[14] M. Lehman, "Short cut multiplication and division in automatic binary digital computers," *Proc. IEE*, vol. 105, pt. 5, pp. 496–504; September, 1958.
[15] I. Y. Akyshky, *et al.*, "Methods of speeding up digital computers," *Proc. Internatl. Conf. on Information Processing*, Paris, France, pp. 365–374; June, 1959, UNESCO-Oldenbourg-Butterworth; Paris-Munich-London; April, 1960.
[16] J. E. Robertson, "A new class of digital division methods," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC7, pp. 218–222; September, 1958.

IX. The Asynchronous Serial Computer

Small serial computers operating with magnetic drums and/or other synchronous memories have, in the past, been synchronized to the store, both in terms of bit rates and word rhythm. Adoption of the time-selection principle for store control permits the machine to operate word-asynchronously with respect to, say, a magnetic drum system. Full synchronization is only required during an actual transfer, and even then not if a two-word buffer is made available. Thus, the machine may be appreciably speeded up, since the whole concept of a unique and fixed word length and word time may be abandoned. For example, storage on a drum may be parity checked through the addition of a parity bit during the transfer process, using a buffer register and possibly a bit rate increased by a factor $n+1/n$. The parity check is then obtained without a corresponding increase in high-speed-store or in the basic word time of the machine.

In general, application of the time-selection principle to the design of the high-speed store enables the adoption of an asynchronous approach to serial-machine design, leading to the abandonment of the classic word-rhythm of serial computers.

X. Conclusions

The paper has presented the concept of a serially read, coincident-current, time-controlled storage system. It has indicated that such systems are economically justified in the smaller stores such as are found in serial, magnetic-drum machines. Advantage may be taken, in the over-all design of the machine, of the fact that individual bits or groups of bits are, at all times, immediately available by the appropriate setting of initial conditions in the time-selection circuits. This leads to an asynchronous design concept resulting in a general speed up of the machine, in increased design and user flexibility and in the simple provision of such refinements as an automatic floating-point facility and high-speed shift, multiplication, and division orders.

Acknowledgment

The author wishes to acknowledge the many useful discussions with and the contributions of the members of the digital computer group of the Scientific Department of the Israel Ministry of Defence.

A Flexible and Inexpensive Method of Monitoring Program Execution in a Digital Computer*

FRANK F. TSUI†, MEMBER, IRE

Summary—A method of monitoring the program execution in a digital computer on the basis of the flow diagram of the computing program has been devised. A comparatively low-cost equipment for monitoring a maximum of 64 boxes in a flow diagram has been constructed.

The monitoring method is flexible and convenient in its application. It can be used in connection with relative or symbolic addresses, compilers, etc. The user must provide only a flow diagram drawn on translucent paper in a certain form and the information to correlate this diagram with the computing program. A subroutine modifies the computing program as needed for the monitoring purpose and restores it to its original form when the user so desires. The monitoring introduces only a very small increase in computing time, requiring for each call-up of a box in the flow diagram only a time amounting to that needed for two simple unconditional jumps. The monitor can be used to present during the computation a visual dynamic picture of the progress of the program and to register, on occurrence, the whereabouts of an interruption, thus facilitating the tracing of the error.

The principle of the monitoring method and the subroutine program, and the essentials of the constructed monitor equipment, are described in detail.

Introduction

THE phenomenally high speed with which an electronic digital computer solves a complicated numerical or logical problem is principally the result of the full utilization of the very high speeds of the electronic circuits under *program-control.* Complicated computations or logical deductions and decisions based on intermediate results can be linked automatically together, and no manual influence is required once the computation is started. This virtual separation of the computer from all "human contacts" during the computation is of essential advantage and is by no means disturbing—so long as the computation proceeds in good order.

Interrruptions in the computation, however, do often occur. These can be due to:

1) Programming errors, which can be:
   a) Errors of a more fundamental nature, *i.e.*, in the logical structure of the program itself. Such errors can often be present in new and "undebugged" manually-compiled programs.
   b) Errors of a conditional nature, dependent on the intermediate results of the computation and not always foreseeable, *e.g.* the overflowing of a certain number-register.
2) Machine failures.

If the execution of the computing program could be followed by means of an external monitor, then, in the case of an undesired interruption, the error could be traced and removed more easily. Such a monitor could also be used to observe the speed of convergence in iterative computation methods, etc.

Theoretically, it is not difficult to monitor program execution in a digital computer. The sequence in which the instructions are carried out is usually directed by an instruction counter (IC) in the control unit, and it is necessary only to observe the content of this register to see which instruction is being carried out at any particular moment. In practice, however, the cost and the convenience and flexibility of application of such a monitor must be considered.

Monitoring of program execution has often been attempted in one form or another. It has been common practice, in case of error, to print out certain intermediate results or parts of the program (*e.g.*, addresses and instructions for certain jumps) as clues to be used in tracing the failure. Experiments have also been done, for instance, on displaying the IC content on a CRO screen, so that jumps and iterative loops with a low repetition rate can be visually followed. These methods, however, have not proven very satisfactory. The former provides only location information and is not very convenient to use; the latter, although possessing the desirable feature of a real-time monitor, displays the location information only in a way which is not directly correlatable to the whole program structure.

Consideration of a New Monitoring Method

The fact that flow-diagram representation continues to be found very helpful by many programmers suggests that the flow diagram could form a suitable basis for a visual monitor. A flow diagram is a graphical portrayal of the course of a problem's solution, later to be approximated by the path of the machine's control through the instruction information stored in the machine during the course of solution. It consists of a sequence of directed line segments or curves extending from one "box" to another, with each box indicating an arithmetic, instructional, or index change, or an assertion about the status of the program.[1] A program monitor that refers to a box in the diagram each time that the operation represented by the box is carried

* Received by the PGEC, October 27, 1960; revised manuscript received, January 25, 1961.

† Institut fuer elektrische Nachrichtentechnik und Messtechnik, Technische Hochschule, Munich, Germany.

[1] "Handbook of Automation, Computation and Control," vol. 2, John Wiley and Sons, Inc., New York, N. Y., p. 2-47; 1959.

out would be effective and synoptical, since it would combine graphical portrayal with the real-time mode of indication.

Technically, such a display should not be difficult to realize. The various operational boxes in the flow diagram could each be marked by a "representative instruction" (RI) chosen from the group of instructions pertaining to the box, and some kind of an indication then could be made on the flow diagram each time this RI is called up and carried out. But such a solution, if executed in the straightforward manner, would be prohibitively expensive.

Flexibility in application requires that the monitor not be bound to any one program or type of program, but should permit the choice of any arbitrary group of RI's and also should preferably be usable with relative or symbolic addressing, compilers, and eventually with automatic programming. A short consideration of an assumed case of, say, N "boxes" being monitored, with the N addresses of their RI's each of A bits being *stored*, to be subsequently scanned for comparison with the IC content (coincidence checking), indicates that such a *storing* method would be either very expensive in circuitry (if the NA bits were to be stored in additional electronic registers provided with gated channels for setting and scanning) or very time consuming in the coincidence checking (if these were to be stored in the common main store). Barring some rather extensive and costly additional hardware, it would be also inconvenient and inflexible to use.

It is seen, therefore, that if an inexpensive and flexible monitor is to be realized on the basis of the flow-diagram suggestion, a method must be found that will:

1) involve the storing of the RI addresses in the main store of the computer, so that the cost of additional registers can be avoided;
2) enable the coincidence checking of these addresses against the IC content, or some other procedure to the same effect, to be carried out in a tolerably short time; and
3) allow the information correlating the boxes and the addresses of their RI's to be fed into the computer in a reasonably convenient and flexible manner, so that the monitor can be used in connection with relative or symbolic addresses, subroutine and compiler programs, etc.

Such a method has been devised and a comparatively low-cost equipment that provides for the monitoring of a maximum of 64 boxes in the flow diagram has been built. These will be described in the following section. It may be mentioned that the monitor, as described below, was built to be used in connection with the PERM (Munich Technische Hochschule Computer), which is a one-address parallel machine with a storage capacity of 10,240 (8192 drum and 2048 core) 51-bit words. With minor modifications, the method should be applicable to other types of computers.

Description of the Monitoring Method and Equipment

Starting from the requirement that the addresses of the RI's to be monitored are, for economy, to be stored in the main store of the computer, the problems now essentially become those of attaining the desired flexibilty and convenience of use of the monitor and of minimizing the time required for coincidence checking. For the former, we can make use of the processing capability of the computer itself. For the latter, the best way is to avoid the time-consuming procedure of coincidence checking altogether. Instead, we can make use of the processing capability, as well as the jumping facilities in the computer, and arrange for a subroutine to modify the computing program after it has been fed in, in such a way that for each of the RI's being monitored, the control will make a detour and "report itself" at a different address, for which logical circuits that will actuate the indication mechanism pertaining to the corresponding flow-chart box are provided. Such a detour, consisting of a simple jump out of and another back to the original control route, would involve a delay amounting to only two jumps, a delay usually tolerable, and in any case small in comparison with that needed for coincidence checking.

With this principle of operation, the monitor equipment should consist of the following:

1) a panel with an assembly of indicator lamps which are arranged in a certain pattern and are allotted position numbers for identification purposes;
2) a certain space in the main store, reserved for the monitoring purpose;
3) a subroutine program for modification and restoration of the computing program;
4) logical circuits for actuating the indicator lamps; and
5) accessories, such as counting circuits for registering, if desired, the number of times certain iterative loops are being run through.

The user of the monitor equipment would be expected to provide the following:

1) A flow diagram drawn on a sheet of *translucent* paper, with the boxes distributed and arranged as required according to the computing program, but occupying positions conformable to the pattern of the indicator lamps on the panel, so that when the diagram is mounted over the panel, the boxes would each be over an indicator lamp; and

2) The information required to correlate the flow diagram with the panel, *i.e.*, the addresses (in absolute,

relative or symbolic form) of the RI's chosen to represent the boxes in the diagram and the corresponding position numbers of the indicator lamps covered by the boxes, which are to be fed in pairs into the computer as a concluding part of the computing program.

The interconnection and operation of these component items will now be described in more detail.

The Panel of Indicator Lamps

Number of Lamps: When binary representation of numbers is used in the computer (as is in the PERM) and the logical circuits for actuating the indicator lamps are to be directed by binary address information through a decoder circuit, the utilization factor (cost per lamp indication) of the decoder would be at an optimum if the number of lamps is a binary power. A loop being traversed can be simulated well by means of three, or at most four, lamps. In most cases the iterative loops of interest are usually coupled inside one another, so that on the average three lamps should suffice. For most programs the flow diagram contains not more than 15 or 20 complicated loops. Hence a total of 64 lamps should be sufficient for most purposes. For some programs perhaps 32 would suffice. An assembly of 64 lamps was therefore decided upon, with the possibility provided for switching half of them out of use to save space in the main store.

Panel Size and Lamp Arrangement: The size of the lamp panel was so chosen that a standard-size (German DIN-A3, 42 cm×30 cm) sheet of tracing paper can be used for the flow diagram. The lamps are arranged in eight rows of eight lamps each, with the neighboring rows staggered with respect to each other to allow a greater freedom of the drawing of iterative loops (see Fig. 1). The lamps used are small neon-glow discharge bulbs. Their positions are numbered 0 to 63, with lamps 0 to 31 arranged in a block that can be covered by a sheet of the DIN-A4 size, 30 cm×21 cm.

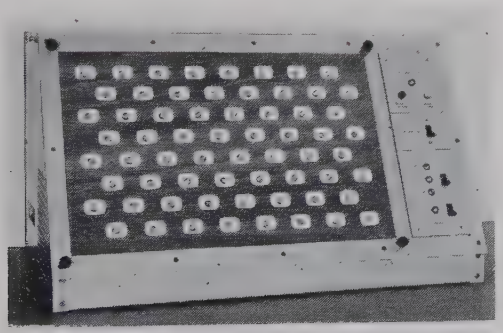


Fig. 1—Front view of the monitor equipment.

The Storage Space Required for Monitoring

The needed space consists of three groups of word-length storage cells used for different purposes:

1) Cells in which the information for correlating the flow diagram with the lamp panel (*i.e.*, the RI addresses and their corresponding lamp-position numbers) will be stored (one cell for each RI). These will be referred to as the "directory list" cells.
2) Cells in which the subroutine for the modification and restoration of the computing program is to be stored.
3) Cells which are to serve as the destination points of the detours (two cells for each RI; this will be explained in more detail later in connection with the description of the subroutine). These will be referred to as the "detour station" cells.

The number of cells needed in 1) depends on the number of RI's to be monitored (*i.e.* of the boxes in the flow diagram); that in 2), on the instruction code used in the computer; and that in 3), (64 or 128 cells) on the size of display field (32 or 64 lamps) chosen. It may be mentioned as an example that, in terms of the code used in the PERM, the subroutine (for modification and restoration) contains about 50 instructions, so that for a display on the 32- or 64- lamp field a total of about 150 or 250 cells respectively are needed for the monitoring. In the following description of the subroutine and the logical circuits, the case of 64 RI's being monitorable generally will be taken. It will be assumed for the present, for the sake of simplicity, that cells with absolute addresses 0 through to 127 are taken as the detour-station cells, and cells 128 to 250 as those for storing the directory list and the subroutine. It will be seen later that the former block of cells may take a different location if required. It is immaterial where the latter groups of cells are situated.

The Subroutine

The subroutine program, which operates only *after* the computing program has been completely fed in, contains the following main functions:

1) To calculate, from the addresses of the first and the last directory-list cells filled by the correlation information fed in, the total number L of lamps to be used for monitoring the given computed program; and, using this L, to prepare for the proper indexed controlling of the iterative loops in the subsequent modification and restoration processes.

2) To modify the computing program as required for the monitoring. The modification consists, for each given RI, of the following operations:

a) A set of the stored RI-address/lamp-number information is read out. It must be explained here that this information *was* given by the user in the form (α_i, n_i), where α_i is the RI address (absolute, relative or symbolic) and n_i the corresponding lamp-number. But if α_i was given in the relative

or symbolic form, it will have been converted to the absolute storage address m_i after the input of the computing program has been completed. It is important that *formally* α_i should occupy in the cell the position for the *operand address*, and n_i that for the *operation-code* part, in the instruction code, so that during the input α_i can be converted into the absolute address m_i in the same way as the operand addresses in the instructions, whereas n_i will not be modified in any way. From the set (m_i, n_i) read out, the address m_i and the lamp number n_i are separated and stored away in temporary auxiliary storage cells.

b) The content of the cell with the address m_i, that is, the ith RI itself, is read out in its entirety and transferred to the cell with the address $2n_i$, which is in the reserved group of detour-station cells.

c) An unconditional-jump instruction: "Jump out to the instruction in cell $2n_i$," is written into cell m_i in replacement of its original content.

d) Another unconditional-jump instruction: "Jump back to the instruction in cell (m_i+1)," is then written into cell $(2n_i+1)$, which is also in the reserved group of detour-station cells.

These operations, shown schematically in Fig. 2, are repeated $(i=1, \cdots, L)$ for all of the L sets of (m_i, n_i). After such a modification, the original linear control route through the cells (m_i-1), m_i and (m_i+1) is now replaced by a detour route through the cells (m_i-1), m_i, $2n_i$, $(2n_i+1)$ and (m_i+1), whereby the call at cell $2n_i$ will furnish the time-saving equivalent to the coincidence checking. Attention is called to the fact that, since the RI's must be transferred away from their original locations m_i to the cells $2n_i$, they should be chosen only from those instructions which are with certainty not to be subjected to any operand-address modifications during the course of the computation.

3) A third and separate part of the subroutine is for the complete restoration of the modified computing program to its original form. This consists of the following operations:

a) A set of the stored RI-address/lamp-number information is read out, from which the m_i and n_i are separated and stored away in temporary auxiliary storage cells.

b) The content of cell $2n_i$, that is, the ith RI itself, is read out in its entirety and transferred back to cell m_i.

c) The cell $2n_i$ is cleared[2] to be ready for use for the next modification process.

[2] Actually, the cell $2n_i$ is restored to contain a starting-condition content, which is a special "indicate the present instruction-address and jump" instruction, to facilitate the calculation of the number L. The cell $(2n_i+1)$ needs no clearing, since its content will be written over by the new jump-back instruction in the next modification.

These operations are repeated $(i=1, \cdots, L)$ for all of the L sets of (m_i, n_i).

With such a modification and restoration subroutine, the following are allowed or possible:

a) The sets of (α_i, n_i) may be fed into the computer in any arbitrary order at the convenience of the user.

b) The monitor can be taken out of operation and the computing program restored to its original form at any instant at will. This may be desirable, for instance, in the testing of new programs, which can be debugged with the aid of the monitor and, when in order, can then be run without monitoring, in order to save computer time or storage space.

c) For a more complicated computing program, the flow diagram may be divided into partial diagrams, so that it can be monitored in greater detail, section by section, if the situation requires. Care should be taken in this case that the RI address for the various partial diagrams should be stored in separate groups and should all be fed in at the beginning, so that the floating-to-absolute address conversion could be performed on all of them beforehand. This would of course require some more storage space and a longer subroutine.

The Logical Circuits

With the computing program modified in the manner described above, all the logical circuits now need to do is to provide for the actuation of lamp number n_i each time its correlated cell $2n_i$ is called up. In principle, the group of the detour-station cells, as space reserved for the transferred RI's and the jump-back instructions, need not be the cells 0 to 127 and, in the extreme case, need not even be in one block at all. But to have them in a block has (besides making the storage-space allocation easier) the important advantage of effectively compressing the probable range of the binary address to be monitored from the full capacity of the computer's storage (usually 2^{12} or more) down to 2^5 or 2^6 (for the case of 32 or 64 lamps), thereby making possible a great simplification of the logical circuits and a considerable saving in the required hardware. It may happen that in some computers the first few cells are constantly occupied for certain purposes. In this case, the block of the 128 detour-station cells (for 64 lamps) may be moved to somewhere else, and the only condition the location of the block must satisfy is that the cells must have the addresses 2^p to (2^p+127), where p is a positive integer subject to the limitation that $7 \leqq p < (A-1)$, where A is the total number of bits in the binary-address representation. In this more generalized case with $p>0$, the lamp numbers n_i are correlated with detour-station addresses (2^p+2n_i), instead of with $2n_i$ as for the case discussed earlier.

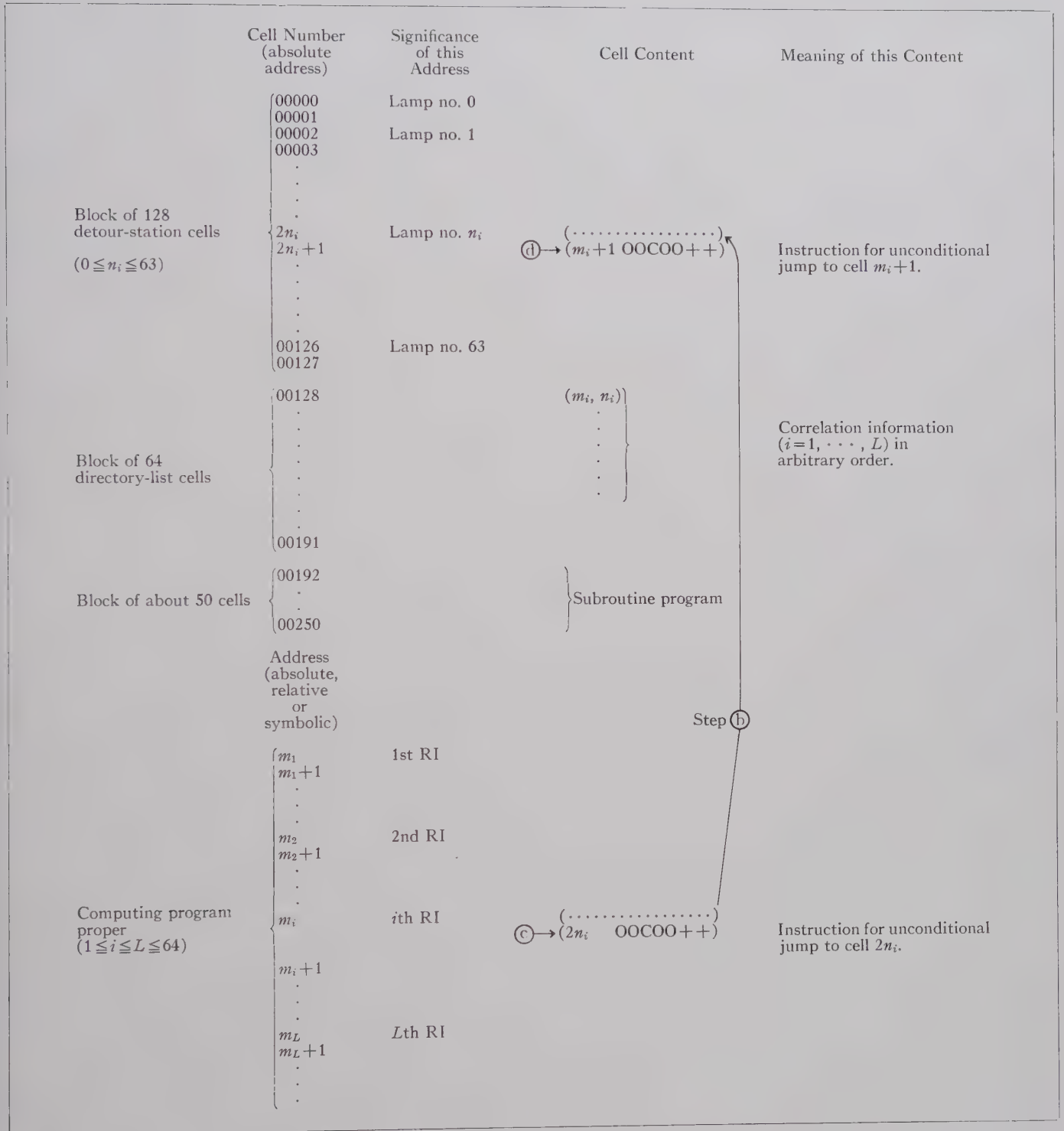


Fig. 2—The main operations contained in the modification subroutine.

With such an arrangement, a possible form of the logical circuits is that as shown in the block diagram of Fig. 3. A "start lamp-display" pulse is combined with the bit-values of the binary places $2^{A-1}, \cdots, 2^{p+1}, \overline{2^p}, 2^{p-1}, \cdots, 2^7$ and 2^0 in the IC through an AND circuit, whose output is combined once again through an assembly of AND circuits, better known as a decoder matrix, with the bit-value pairs of the remaining 6 binary places 2^6 to 2^1 and $\overline{2^6}$ to $\overline{2^1}$. Each of the 64 outputs of this matrix are then connected to a voltage-controlled flip-flop circuit and an indicator lamp. The flip-flops may further be connected by selection to counting circuits for registering the numbers of callups, if desired. The component parts of this circuitry are described in some detail in the following.

1) The "*start lamp-display*" *pulse* is concurrent with (and should be no earlier than[3]) the trigger pulse that initiates the actual instruction-*execution* in the computer proper.

2) The first *AND circuit* that combines this start-pulse with the binary places $2^{A-1}, \cdots, \overline{2^p}, \cdots, 2^7$ and 2^0 is easily realized with (A-5) diodes. Its output is reinforced with a cathode-follower before going into the *decoder matrix*, which is to combine 13 inputs (the AND output above the 6 flip-flop output pairs) into 64 output channels. In the straightforward structure-form $(s \cdot 2^6 \cdot 2^5 \cdot 2^4 \cdot 2^3 \cdot 2^2 \cdot 2^1)$, the matrix would require $2^6(1+6)$ or 448 diodes. One optimum structure, however, can be shown to be of the form $(s \cdot 2^6 \cdot 2^5) \cdot (2^4 \cdot 2^3) \cdot (2^2 \cdot 2^1)$, which requires $2^2 \cdot 3(1+2^2+2^4)$ or 252 diodes (see Fig. 4).[4] It is not necessary for this matrix or the preceding AND circuit to have very fast rise and fall times. The highest operating frequency for these AND circuits is that with which the IC content may be changed, and it is in most cases one or two orders of magnitudes slower than the clock frequency in the computer itself. With this milder requirement in the transient response, the cascaded-AND construction of the matrix can be realized with only a few interstage cathode-followers necessary.

3) As *indicator lamps*, small neon bulbs are used. These have a firing voltage of about 80 v and an extinguishing voltage of about 60 v, both with considerable scatter. Since the decoder matrix has rather high-impedance output channels and its output pulse has only a limited duration[5] which is very much shorter than the time a neon bulb needs to produce a visible flash, some kind of an amplifying and pulse-stretching stage must be inserted between the matrix output and the lamp. For this purpose, the *flip-flop* circuit was chosen in preference to other possible alternatives, since, in addition to providing the amplification needed, it can possess the property either of a delaying element (in the monostable circuit-form) or of a storage element (in the bistable form). It is desirable to unite both of these in one element so that, while during the normal operation of the computer the monitor can give transient flashes in the different lamps with a duration determined by the delaying property in the monostable form, the storage property in the bistable form can be used to produce a persistent display in the last actuated lamp or lamps on the occurrence of an interruption. Such a mono/bistable stage, whose circuit form can be controlled by an external control voltage V_c, is shown in Fig. 5. Sixty-four such stages,[6] one for each output channel of the decoder

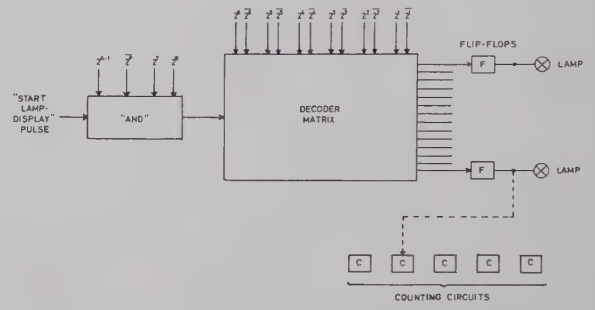


Fig. 3—Block diagram of the logical circuits of the monitor equipment.

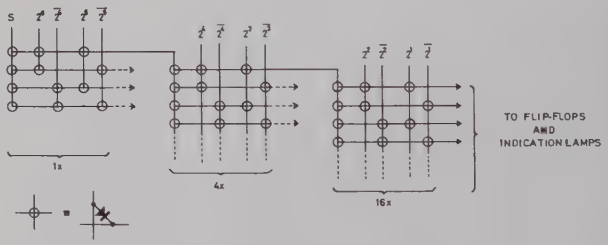


Fig. 4—Structure of the diode decoder matrix.

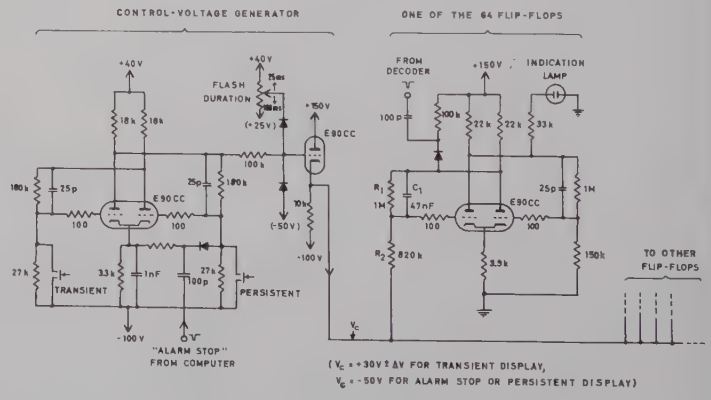


Fig. 5—Circuit diagram of the control-voltage generator and the voltage-controlled flip-flop.

[3] This is especially important in cases where the control in the computer works with the time-sharing principle of advance preparatory interpretation of instructions. Otherwise, spurious display may occur through changes of the IC content in case of jumps, etc.

[4] This structure form has, furthermore, the advantage that the whole matrix may be divided into smaller identical subunits, so that the printed-circuit technique can be used.

[5] This is because the "start lamp-display" pulse cannot be made longer than the shortest probable interval between the start of an instruction execution and the subsequent changing (counting-up-by-one, etc.) of the IC content.

[6] Since it is also not necessary for these flip-flops to have very fast rise and fall times, they can be built with aged tubes rejected from the computer (if it is one with tubes) because of reduced cathode emission.

matrix, are used for the actuation of the 64 lamps in the display panel.

4) The *control voltage* V_c is generated by another flip-flop circuit, which is bistable and whose two anode-output levels lie positive and negative with respect to the cathodes of the lamp-actuation flip-flops. Its output is fed through a limiter circuit to make the positive level adjustable and, after being reinforced by a cathode-follower, is used as the control voltage for all the 64 lamp-actuation flip-flops.

By changing the state of the V_c-generator flip-flop, the lamp-actuation flip-flops can be made monostable or bistable, thereby providing the monitor display with the following useful features:

1) The duration of the transient flashes in the indicator lamps can be varied to obtain the best optical effect. This is done for all of the 64 lamps in common by varying the V_c in its positive limited-level value. Measurements on neon bulbs have indicated that the quasi-stable duration of the monostables should be 20 msec or longer if the bulbs were to be allowed to reach their maximum brightness. An adjustment for the flash duration to be variable between 25 msec and 100 msec therefore seems desirable because, under the normal conditions of operation, the RI's would not be chosen very close to one another, and a moderate overlapping of the flashing times would rather enhance the optical effect of a light point in motion.

2) On occurrence of an undesired interruption in the computation, a reliable registration of the whereabouts of the interruption can be obtained by arranging for the V_c-generator flip-flop to be "set" by an "alarm and stop" pulse from the computer proper. This turns V_c negative and makes all the actuation flip-flops bistable, thus leaving the last one or two lamps lighted immediately before the interruption "on."

3) By providing for a manual setting and resetting of the V_c-generator flip-flop, the display can be changed at will from the transient mode to the persistent, and vice versa. The persistent mode of display may be used, for instance, to register which parts of the computing program have been run through (conditionally selected program parts or subroutines, etc.) or to serve as a direct indicator of the scope of the computation or of the results of logical deductions, etc.

These arrangements can be seen from the circuit diagram in Fig. 5. A view of the complete monitor equipment with the whole assembly of logical circuits and flip-flops built behind the lamp-panel is shown in Fig. 1.

Conclusion

The paper has described a method of monitoring the program execution in a digital computer in terms of an indicator-lamp display on the flow diagram pertaining to the computing program, has discussed the essential parts of a monitor equipment constructed on the basis of this method, and has explained in detail the subroutine program which takes care of the necessary preparations for the monitoring. It has shown that the monitor is flexible and convenient to use, introduces only a very small increase in computing time, and can be built inexpensively. The monitor has, furthermore, the desirable features of presenting during the computation a visual dynamic picture of the progress of the program and of registering on occurrence of an interruption the whereabouts of this interruption to facilitate the tracing of the error.

While the monitoring method and equipment, as described, can be used in connection with relative and symbolic addressing, compilers, etc., it is believed that its application can be extended without much difficulty to programs compiled through automatic programming and ALGOL formulations, if adequate provisions can be made, in the input-language code and the translator subroutine, for the proper earmarking of the operation steps to be monitored. The possibilities in this direction are being investigated and will be tried out.

Acknowledgment

The author wishes to express his thanks to K. Voit for his assistance in the construction of the monitor equipment and to W. Anacker for many helpful discussions. Thanks are also due to Dr. H. Piloty for his interest and support of this work and for his permission for the publication of this paper.

On the Encoding of Arbitrary Geometric Configurations*

HERBERT FREEMAN†, SENIOR MEMBER, IRE

Summary—A method is described which permits the encoding of arbitrary geometric configurations so as to facilitate their analysis and manipulation by means of a digital computer. It is shown that one can determine through the use of relatively simple numerical techniques whether a given arbitrary plane curve is open or closed, whether it is singly or multiply connected, and what area it encloses. Further, one can cause a given figure to be expanded, contracted, elongated, or rotated by an arbitrary amount. It is shown that there are a number of ways of encoding arbitrary geometric curves to facilitate such manipulations, each having its own particular advantages and disadvantages. One method, the so-called rectangular-array type of encoding, is discussed in detail. In this method the slope function is quantized into a set of eight standard slopes. This particular representation is one of the simplest and one that is most readily utilized with present-day computing and display equipment.

Introduction

CONCOMITANT with present-day advances in the fields of control and communications, a growing need is being felt for precise means of describing arbitrary geometric configurations. This need arises in two ways: 1) from the desire to communicate geometric information, and 2) from the requirement for analysis and synthesis of geometric configurations.

If one wishes to transmit the outline of a lake, the information relating to the contour must first be identified (a description process) and then encoded in a manner which permits efficient transmission. Decoding and reconstruction of the contour in terms of a pictorial representation is required at the receiving end of the transmission. The twin objectives of communication are efficient transmission and faithful reproduction. Any scheme for communicating geometric information must be able to withstand critical examination on these objectives.

The analysis and synthesis of arbitrary geometric configurations have, until recently, received only scant attention; the tedium of lengthy hand computation was too discouraging. With the availability of modern high-speed digital computers, however, labor of computation is no longer a deterrent. Thus it is now entirely feasible to use a computer to determine the surface area of a lake from its contour, or, given a set of past weather maps, to compute a most probable future weather map. One need merely develop the necessary algorithms for the computer and devise methods for encoding the geometric information in a way which will facilitate such analysis.

* Received by the PGEC, August 8, 1960; revised manuscript received, January 27, 1961.

† Dept. of Elec. Engrg., New York University, New York, N. Y.

An arbitrary geometric curve is a curve which is pictorially defined (*e.g.*, the outline of a knife), but for which no analytic description exists. It thus differs from a mathematical (or nonarbitrary) curve, such as a triangle, sine wave, or hypocycloid, which can be precisely formulated in mathematical terms. A circle is a mathematical curve perfectly described by a simple equation. In contrast, a wheel is an arbitrary geometric curve which, to a certain limited degree of precision, can be approximated (described) by a circle. If one desires descriptions which will take into account more and more of the wheel's minute irregularities, the descriptions must become progressively more complex and unwieldy.

Description Processes

Consider the problem of describing arbitrary geometric curves to permit their transmission over a telegraph communication link or their manipulation by a digital computer. The curves are assumed to be continuous plane curves, open or closed, and either singly or multiply connected. One approach to this problem is to cut the curve into arbitrarily fine segments and then to approximate each segment with a standardized mathematical curve.

The segmentation of curves raises some important questions. Should the segments be large and few in number, or should they be small and much greater in number? If the former, the mathematical representation of the segments will be relatively complex (*e.g.*, high-order polynomials may be required), while, if the latter, a straight-line approximation will suffice. Further, should the segments all be equal or should segmentation be dependent on the curve itself? Uniform segmentation has the advantage of simplicity; however, segmentation which is based on the properties of a particular curve may lead to more efficient encoding.

An encoding scheme for arbitrary geometric curves should be simple, highly standardized, and universally applicable to all continuous curves. Further, the encoding scheme should be such that it facilitates digital computer analysis of curve properties. For these reasons subsequent discussion will be limited to straight-line segmentation using either a single length or a small number of standard lengths. It is not intended to imply that other approaches are without merit; their consideration will, however, be outside the scope of this paper.

A question that arises at once concerns the fineness to which a curve is to be segmented. Consider the curve of Fig. 1, drawn in the x-y plane. One may introduce a parameter t which will be a measure of distance along the curve as the curve is traversed in the direction indicated, beginning at the starting point x_0y_0. This makes it possible to represent the original curve by two parametric curves, $x(t)$ and $y(t)$, as shown in Fig. 2.

Approximating a plane curve with straight-line segments is a sampling-type operation, and its validity can be tested by means of the sampling theorem. This can be done by obtaining the Fourier transforms of the two parametric curves and examining their frequency spectra. Generally, the frequency spectra will be infinite, with progressively increasing attenuation of the higher-frequency components. A qualitative interpretation of the sampling theorem states that the sampling interval T must be less than one half the reciprocal of the highest significant frequency component. If the precision of representation is to be such that over a square-meter x-y plane any errors in location of points on the curve will not exceed one millimeter, one may refer to the representation as having a precision of 60 db. In a loose sense this can be carried over into the frequency spectrum, where components which are attenuated 60 db or more may then be considered insignificant.

Once the desired segment-length T has been determined, one may replace the original arbitrary curve with a curve consisting of a chain of straight-line segments of length T. The curve can then be described simply by a sequence of angles and a standard distance. When reconstructing the curve from the given data, the scale can be arbitrarily selected by choosing the appropriate standard length. A coded representation of

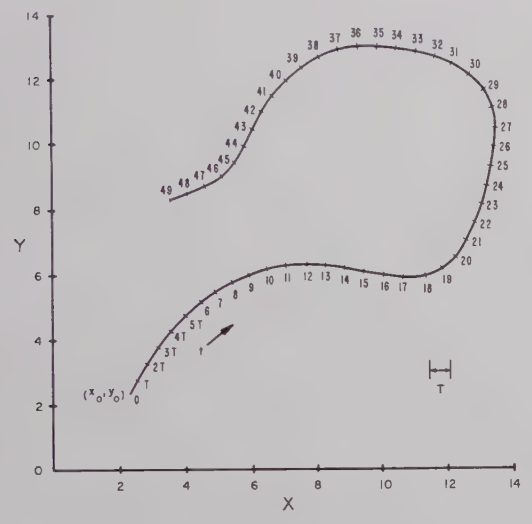


Fig. 1—Arbitrary plane curve.

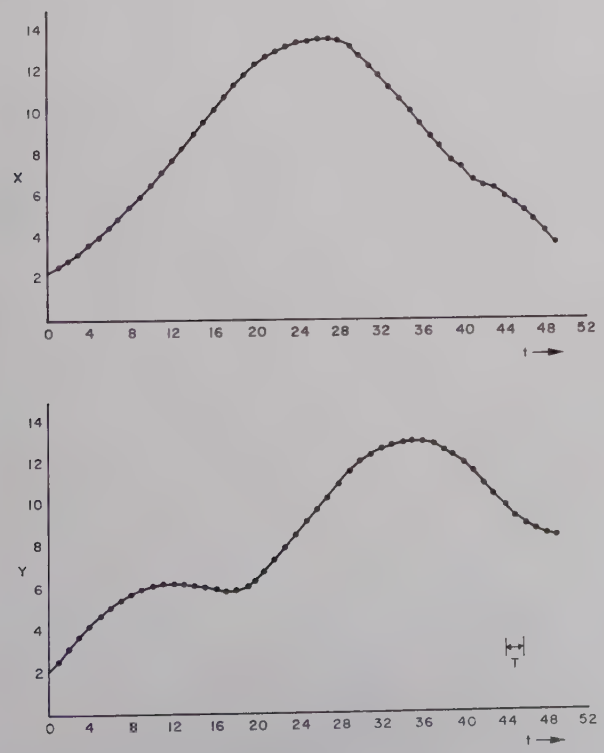


Fig. 2—Parametric curves.

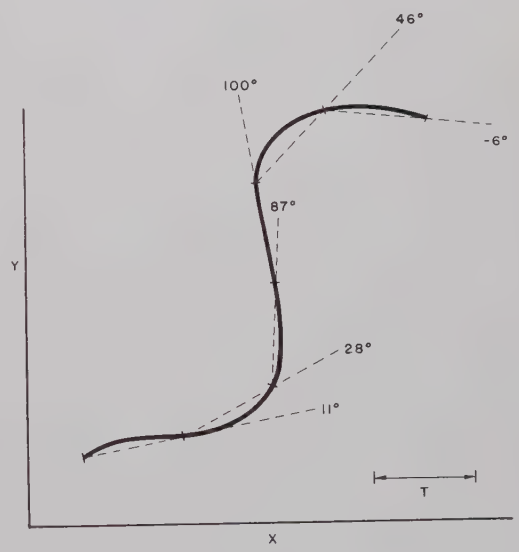


Fig. 3—Coarsely quantized arbitrary plane curve.

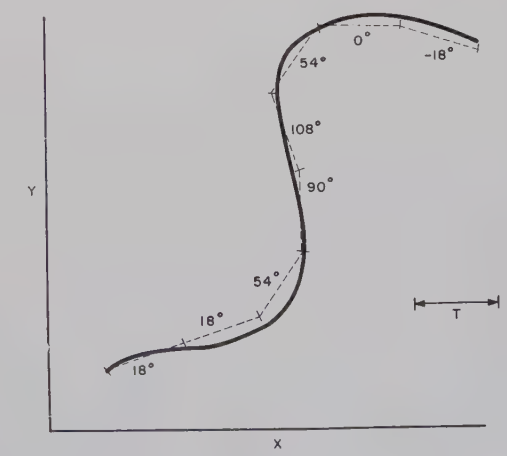


Fig. 4—Quantization based on a standard angle of 18°.

the curve shown in Fig. 3 would appear as follows:

$$11°, 28°, 87°, 100°, 46°, -6°.$$

The actual encoding procedure can be simplified by severely limiting the number of permissible angles. Thus if one were to limit the encoding to 20 standard angles (18° increments) and to number these consecutively from 0 to 19, the curve of Fig. 4 could be represented by the following number series (read from left to right):

$$1, 1, 3, 5, 6, 3, 0, 19.$$

It is evident that any desired precision (fineness of resolution) can be obtained with a relatively coarse angle quantization by making the standard distance arbitrarily small.

This method of describing an arbitrary geometric curve may be termed the unit-vector method. In essence, a curve is represented by a sequence of small vectors of unit length and a limited set of possible directions. For reasons of data handling, it is sometimes preferable to use a grid or array type of description. Thus, one may lay a rectangular grid over the curve and describe the curve by identifying the grid points which lie closest to it. Such a grid system has the advantage over the unit vector scheme of being much less susceptible to error build-up during encoding.

Rectangular-grid representations of arbitrary geometric curves are frequently encountered in connection with digital-computer output displays. Thus, a common type of display consists of a cathode-ray tube on the face of which the beam spot can be positioned to, say, one part in 1024 in both the x and the y direction. Two ten-place binary numbers (20 bits) are required to locate a particular point on the face of the tube. Although a typical curve display may require a very large number of points, the full 20-bit coordinates of each point are usually given. This is unnecessarily wasteful.

The rectangular grid method is somewhat similar to a unit-vector method based on eight angles. The significant difference lies in the use of two standard distances which are in the ratio of $\sqrt{2}$ to each other; *i.e.*, one is the side and the other is the diagonal of a small square in the grid.

The unit-vector representation for six angles has properties that merit special consideration; it can also be regarded as being a representation based on a hexagonal grid. This is the only grid utilizing a single standard distance. The use of a hexagonal grid for computer output display can be achieved with a cathode-ray tube having its pairs of deflection plates mounted at 60° to each other instead of the conventional 90°.

Coding Processes

The process of adopting a symbolism convention and then quantitatively identifying the describing properties in terms of the convention is called encoding. It is desired here to develop means for encoding arbitrary geometric curves so as to facilitate their handling in communications systems and digital computers. A general objective of all encoding is that the process be efficient, *i.e.*, that it use a minimum of bits to represent some given information.

Consider the rectangular-grid representation mentioned previously. The conventional manner of representing a curve consists of giving the full twenty-bit coordinates of each point of a curve (for a 1024×1024 array). Such coding would be necessary if all points were fully independent of each other. If the points describe a continuous curve, they are, of course, far from independent. An important restriction which holds for continuous curves (and which may be considered a definition of a continuous curve) is that successive points are adjacent to each other. Once one point on a continuous curve is given, the next point's location is very much restricted. This knowledge can be utilized to locate successive points with far fewer bits than the number required if each point were to be independently located.

For the rectangular grid it is clear that if a point on a continuous curve is known, the next point can assume only one of eight possible adjacent positions, as shown in Fig. 5(a). If one assigns the decimal digits zero through seven to these eight positions, starting with the one which is horizontally to the right and progressing in a counter-clockwise direction, the code of Fig. 5(b) is obtained. One notes that in this manner each point on a curve is located relative to the previous one, and only the first need be located in an absolute sense. To shift a curve vertically or horizontally, only the coordinates of the first point require changing. As an illustration, consider the curve of Fig. 6 which can be represented by the code sequence (read from left to right).

$$107643,$$

or in binary form,

$$001 \quad 000 \quad 111 \quad 110 \quad 100 \quad 011.$$

One notes that only three bits are required to specify one point on the curve. The required memory capacity for a continuous curve encoded in this way is then only 15 per cent of that required for a curve which has all its points independently specified in a 1024×1024 point array.

One property of the rectangular-array code is at once apparent. Adding "2" (modulo 8) to each of the digits in the decimal representation of a curve causes the curve to be rotated 90° in a counter-clockwise direction. A subtraction will result in clockwise rotation. The addition (or subtraction) of any even number n will cause a rotation equal to $n/2$ times 90°. If "1" is added to each digit, a rotation of 45° is achieved; however, this rotation is nearly always accompanied by distortion or at least a change of scale. This is true for the addition (or subtraction) of any odd number.

The encoding method discussed thus far is by no means unique. Nor is it necessarily the one which is most

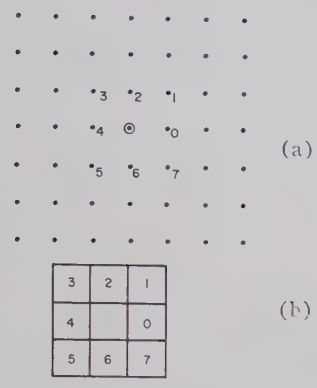


Fig. 5—Rectangular array.

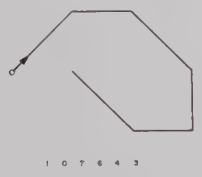


Fig. 6—Illustration of rectangular-array representation.

economical in its use of memory capacity. In developing the method it was supposed that, given one point, the next point could assume one of eight neighboring positions with equal likelihood. If certain conditions are imposed on an arbitrary curve (in addition to continuity), a two-bit or even a one-bit code can be utilized. For example, if it is known that the slope of the arbitrary curve changes very slowly, *i.e.*, if the values of dx/dt and dy/dt at $t=n$ differ only slightly from those at $t=n+1$, then a one-bit code may suffice. Such a code would be associated with the second difference, and would be used as follows. Given the slope between the previous two points, the slope to the next point would be formed by increasing the last slope by 45° if the digit is "1," or decreasing the slope by 45° if the digit is "0." A straight line would be represented by a sequence of alternating ones and zeros.

The rectangular-array code of Fig. 5 has the disadvantage, as has been noted, that the addition of an odd number to each digit results in distorted rotation. This drawback can be avoided by using instead the hexagonal array shown in Fig. 7(a). For each given point, there are six possible locations for the next point, all of which are equidistant from the given point. The six next-point locations can be numbered from zero to five in a counter-clockwise direction, starting with the location horizontally to the right. Three bits per digit are required to represent the six next-point locations. The code is unsaturated since there are two unused combinations, and special techniques must be used to obtain modulo-six additions. An example of a hexagonal-array display is given in Fig. 8. The curve is represented by the sequence (read from left to right):

10011554343.

The hexagonal-array representation has the property that addition (modulo six) of unity to each digit causes

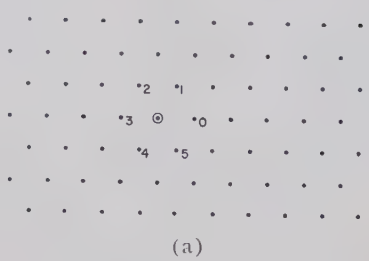


Fig. 7—Hexagonal array.

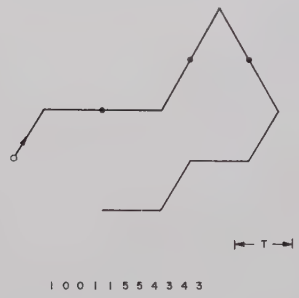


Fig. 8—Illustration of hexagonal-array representation.

a 60° distortion-free rotation of the curve. The method suffers from the disadvantage of being based on an uncommon, nonorthogonal coordinate system, of having its angular resolution limited to 60° (in contrast to 45° for the rectangular array), and of resulting in an unsaturated binary code representation.

The simplicity of the rectangular array, as well as the ease with which it can be adapted to standard display techniques, suggests that this array be given special attention. Accordingly, the remaining sections of this paper are devoted entirely to this encoding method. It is usually not difficult to see, however, that most techniques applicable to the rectangular array can be extended to the hexagonal and the unit-vector arrays.

Rectangular-Array Encoding

In the rectangular-array method, based on the code of Fig. 5(b), a number sequence utilizing only one kind of digit represents a straight line; *e.g.*, the sequence 22222222 · · · 2 describes a vertical straight line. A total of eight such straight lines can be represented in this manner with the digits "0" through "7." Each of these lines makes an angle with the horizontal which is equal to the digit multiplied by 45°. Straight lines making angles with the horizontal of other than multiples of 45° can be formed with digit pairs, triplets, or higher-order digit groups. Thus the sequence 121212121212 · · · 12 represents a "straight" line making an angle of +63.43°. The total number of straight lines which can be represented with digit pairs is 16. For triplets this increases to 32. One notes that these digit groups can contain only

two kinds of digits, and that these cannot differ by more than unity, *e.g.*, 0 and 1, 1 and 2, etc.

Examples of a variety of straight-line representations are given in Fig. 9.

All first-quadrant straight-line digit groups up to five digits in length are listed in Table I. Angles in other quadrants can be obtained by adding either 2, 4 or 6 to each of the digits in a group.

The digit groups given in Table I are not necessarily unique. Thus, the digit groups 01010 and 00011 are equivalent; *i.e.*, they describe the same straight-line slope. However, the first digit group is to be preferred because of its greater smoothness. To assure minimum deviation from the true straight-line path, the two kinds of digits making up a digit group should alternate to the maximum extent.

The representation of an arbitrary curve of any significance will usually consist of a sequence of many hundreds of digits. The more precisely an arbitrary curve is to be represented, the more finely it must be quantized, and, therefore, the greater the number of digits required in its representation. If a curve is completely arbitrary (except for continuity), no condensation or abbreviated representation is possible. However, if there is some regularity in the curve, this *pattern* can be exploited to simplify the representation.

Consider, for example, the straight line which makes an angle of $+26.57°$ with the horizontal. It is represented by the sequence

$$0101010101010101 \cdots 01.$$

One may regard this number sequence as a function of a parameter n where $n = 0, 1, 2, \cdots, N$. A more compact representation is then possible through the introduction of the concept of the generating function.[1]

Given a function $f(n)$, one defines another function $u(x)$ as follows:

$$u(x) = \sum_{n=0}^{N} f(n)x^n,$$

where $u(x)$ is the generating function of $f(n)$. For the straight line just mentioned, the generating function is given by the following:

$$(0) + (1)x + (0)x^2 + (1)x^3 + (0)x^4 + (1)x^5 + \cdots.$$

This infinite sequence has the closed form

$$\frac{x}{1 - x^2}.$$

In obtaining $f(n)$ from the corresponding generating function, one must recognize that a zero is a meaningful digit which represents a finite segment of the curve being described. Generating functions corresponding to the first eight entries of Table I are given in Table II.

[1] C. Jordan, "Calculus of Finite Differences," Chelsea Publishing Co., New York, N. Y., p. 20; 1950.

Fig. 9—Straight-line representations.

TABLE I

First-Quadrant, Straight-Line Digit Groups

Digit Group	Tangent	Angle (Degrees)
0	0	0
00001	0.2	11.32
0001	0.25	14.04
001	0.33	18.4
01010	0.4	21.8
01	0.5	26.57
10101	0.6	30.09
110	0.67	33.7
1110	0.75	36.85
11110	0.8	38.65
1	1.0	45
11112	1.25	51.3
1112	1.33	53.1
112	1.5	56.3
12121	1.67	59.01
21	2	63.4
21212	2.5	68.2
221	3	71.6
2221	4	75.97
22221	5	78.7
2	∞	90

TABLE II

	Digit Group	$u(X)$
1	0	$\frac{0}{1 - X}$
2	00001	$\frac{X^4}{1 - X^5}$
3	0001	$\frac{X^3}{1 - X^4}$
4	001	$\frac{X^2}{1 - X^3}$
5	01010	$\frac{X + X^3}{1 - X^5}$
6	01	$\frac{X}{1 - X^2}$
7	10101	$\frac{1 + X^2 + X^4}{1 - X^5}$
8	110	$\frac{1 + X}{1 - X^3}$

The concept of generating function representation is not limited to straight lines. Consider the sequence 01234567. This sequence will be called the minimal circle and is shown in Fig. 10. It is the smallest figure which possesses the circle property of permitting arbitrary rotation without any visible effect. It is minimal in that there is no smaller figure for which this property holds.

The minimal circle is a closed curve and is represented by a short sequence of digits. Even when dealing with short sequences, it is sometimes desirable to employ the generating function representation. For the minimal circle the generating function is given by

$$\frac{x}{(1-x)^2},$$

as is easily verified by performing the indicated long division. This generating function leads to an infinite series; however, since the numbers are all modulo eight, the series is periodic and simply causes the same path to be traversed over and over again.

Various minimal waveforms, such as square waves or triangular waves, can be represented in a straightforward manner. Minimal waveforms are waveforms which cannot be reduced without destroying the characteristics of the waveform. A set of common minimal waveforms, together with their digit groups and generating functions, is shown in Fig. 11. The numerators

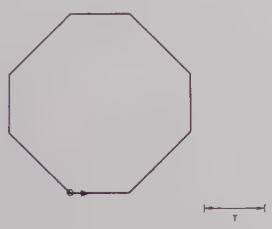


Fig. 10—Minimal circle 01234567.

NAME	WAVEFORM	DIGIT GROUP	GENERATING FUNCTION
SQUARE		0206	$\frac{2x+6x^3}{1-x^4}$
SAWTOOTH		16	$\frac{1+6x}{1-x^2}$
TRIANGULAR		17	$\frac{1+7x}{1-x^2}$
SINE		121767	$\frac{1+2x+x^2+7x^3+6x^4+7x^5}{1-x^6}$
RECTANGULAR	a b	20...0 6 0...0 (a, b)	$\frac{2+6x^{a+1}}{1-x^{a+b+2}}$
BACKWARD SAWTOOTH		27	$\frac{2+7x}{1-x^2}$
SPACED SAWTOOTH		160	$\frac{1+6x}{1-x^3}$

Fig. 11. Minimal waveforms.

of the generating functions bear a one-to-one correspondence to the digits of the respective digit groups, *i.e.*, to the digits in one period of the sequences. The denominators are determined by the period lengths of the sequences. Thus, if a sequence has a period of three digits, the denominator of the corresponding generating function becomes $1-x^3$.

Elementary Manipulations

One of the objectives of encoding an arbitrary geometric curve is to translate the significant information contained in the curve into a representation which will facilitate manipulation of the curve by a digital computer. The problem may be one of analysis, *i.e.*, to discern certain properties of a curve such as its area or axes of symmetry, or it may be one of synthesis, where a curve is to be constructed having specified properties. The following section will describe some of the simpler techniques for manipulating curves encoded in accordance with the three-bit rectangular array.

Expansion

Consider a curve stored in the memory of a digital computer. If displayed by the computer's graphical output unit (CRT or similar device), the curve will have an over-all size determined jointly by the coded representation of the curve and the scale factor of the display device. If it is desired to expand a portion of a curve relative to the rest of the curve, an operation must be performed on the coded representation of the curve. Changing the scale factor of the display device would have a uniform effect and could not be utilized for selective expansion.

The process of expansion is performed as illustrated by the following example. Given a curve represented by 012075, a curve exactly twice this size, but otherwise indistinguishable, is given by 001122007755. To expand a curve by a ratio n, each of the digits of the curve must be replaced by a set of n identical digits. One notes that n must be an integer.

If a curve is represented by a generating function, expansion can be performed analytically by replacing x by x^n and then multiplying by $(1-x^n)/(1-x)$. Thus the minimal circle 01234567 can be tripled in size by performing the appropriate operation on its generating function:

minimal form:

$$\frac{x}{(1-x)^2},$$

expansion to triple size:

$$\frac{x^3}{(1-x^3)^2}\,\frac{1-x^3}{1-x}=\frac{x^3}{1-x-x^3+x^4}.$$

Contraction

Over-all contraction is readily accomplished by an appropriate change of scale; however, selective contraction requires a special procedure and is almost al-

ways accompanied by distortion. This distortion is introduced because contraction implies (except for trivial cases) a reverting to a more primitive representation, and, consequently, a loss of information. Consider the contraction to one-half size of the curve 22101076. If the digits could be grouped into pairs of identical digits, the contracted curve would be given merely by selecting one digit from each pair. The first two digits are identical and meet this requirement. All the remaining digits, however, do not occur in pairs. One notices that the next four digits can be arranged to give two pairs, "11" and "00." The last two digits differ by unity and their mean does not favor any integer. In such a case, the previous digit is also considered and a mean of these three digits is obtained. The mean for 0 (*i.e.*, 8), 7, and 6 is the digit 7. The contraction is thus as follows:

22101076
22110076
22110077
2107.

Rotation

Rotation has already been discussed. Where the rotation is in multiples of 90°, one need merely add the appropriate even digit to each digit of the curve. Rotations of odd multiples of 45° are obtained by adding the appropriate odd digit. Such rotations are almost always accompanied by distortion. Rotations of arbitrary amounts involve digit-group substitutions and are beyond the scope of the present discussion.[2]

Curve Length

The length of a curve is given by the number of even digits plus the square root of two times the number of odd digits.

Inverse Digits

Inspection of the array of Fig. 5(b) shows that digits which are diametrically opposite in the array have a canceling effect on each other. Accordingly, such digits are referred to as inverses of each other. Thus the inverse of "1" is "5," etc. Numerically, the inverse of a digit is obtained simply by adding either plus or minus "4" to the digit—whichever gives a positive sum less than "8" (hence, adding "4" modulo 8).

Path Reversal

A curve is defined by selecting a starting point and giving a sequence of slope segments which trace out the path of the curve from this point. Path reversal is the process of tracing out a curve in reverse direction. It is achieved by replacing each digit of a curve sequence by its inverse and reversing the sequence end for end. As an example, path reversal changes the curve sequence 1232076 to 2346765.

Path Reduction

Path reduction is defined as the process of obtaining the shortest distance between the end points of an open curve and is found by reducing the given curve in the following manner. If a curve contains two digits which are inverses of each other, they may both be deleted without affecting the location of the end points. Further, there are many digit pairs which can be replaced by a single equivalent digit. Thus, the digit pair "42" can be replaced by the single digit "3." The two digits need not be adjacent for either a cancellation or a replacement to be possible. Table III lists all possible sum-replacement and inverse-digit pairs. There are some digit combinations which, although not permitting reduction, can be replaced by a pair of identical digits. Such digit combinations are presented by the double-digit entries in Table III.

TABLE III

	0	1	2	3	4	5	6	7
0			1	2	x	6	7	
1				22	2	x	0	00
2	1				3	4	x	0
3	2	22				44	4	x
4	x	2	3				5	6
5	6	x	4	44				66
6	7	0	x	4	5			
7		00	0	x	6	66		

x = Inverse

As an illustration, let it be desired to find the shortest curve connecting the end points of the curve 7107542. The length of this curve is $3+4\sqrt{2}$. The curve has two cancellation pairs, "15" and "04," which gives a partially reduced curve 772. Sum-replacement reduces this further to the two-digit curve 70. The minimum distance between the curve's end points is thus equal to $1+\sqrt{2}$.

Closure

To determine whether a given curve is open or closed, one performs a path reduction. If the curve vanishes, it is closed. If it does not vanish, it is open, and the residue (referred to as the closure) is the minimum distance between the end points.

As a corollary to the above, one notes that the digits of a given curve may be arbitrarily rearranged without changing the length of the curve or the minimum distance between its end points.

[2] For example, one notes from Table I that rotating the straight line which makes an angle of 26.57° with the horizontal so that its angle will be 30.09°, requires replacement of the digit group 01 by the digit group 10101.

Multiple-Loop Curves

To determine whether a given curve consists of multiple loops, one must examine the curve for subsets of adjacent digits for which the closure vanishes. Thus the curve 7012345667012345 has a total of seven subsets (including one for the curve itself), for which the closure vanishes, indicating the existence of seven possible ways of tracing out a closed path within the curve; however, only three of these loops are independent.

If a curve has more than one subset for which the closure vanishes, and if the subsets are nonoverlapping and separated by an ordered set of cancellation pairs, the curve is multiply connected except that the customary connecting branch has been provided (represented by the ordered set of cancellation pairs). The curve 01123345567723210765 46 is an example of a multiply connected curve. Both this and the previous curve are illustrated in Fig. 12.

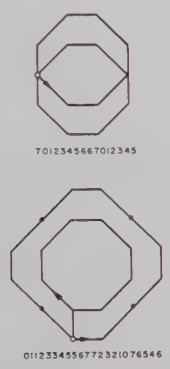


Fig. 12—Multiple-loop curves.

Intersection

To find the point of intersection of two curves, one must first specify a connecting branch which will place the two starting points into relative position. The connecting branch may be any simple curve tracing out a path from the starting point of one curve to the starting point of the other. It may not, however, touch the two curves except at their starting points. Tracing out first one curve and then the other (from the starting point of the connecting branch), a progressive path reduction is performed by considering one additional digit at a time. Each reduced curve is rearranged to place its digits in numerically ascending order. The corresponding lengths of the reduced curves are also computed. The two sets of path reductions are then compared. For each path-reduction sequence which is *common to both curves*, a nodal intersection exists (*i.e.*, the curves intersect at a grid point). If a comparison of the path-reduction sequences of the two curves shows them to differ by only one digit (either by a difference in value of one or the absence of one digit) twice in succession, and if simultaneously the corresponding lengths of the sequences interchange their roles as to which is the longer, a non-nodal intersection exists (*i.e.*, an intersection between grid points).

As an example, consider the two curves $A = 01066112$ and $B = 020117$, connected by a branch 66 from A to B. The successive path-reduction sequences and lengths are as follows:

		Sequence	*Length*
For A:	1)	0	1
	2)	01	$1+\sqrt{2}$
	3)	$010 = 001$	$2+\sqrt{2}$
	4)	$0016 = 000$	3
	5)	$0006 = 007$	$2+\sqrt{2}$
	6)	$0071 = 0000$	4
	7)	00001	$4+\sqrt{2}$
	8)	$000012 = 00011$	$3+2\sqrt{2}$
For B:	1)	6	1
	2)	66	2
	3)	$660 = 67$	$1+\sqrt{2}$
	4)	$672 = 7$	$\sqrt{2}$
	5)	$70 = 07$	$1+\sqrt{2}$
	6)	$071 = 000$	3
	7)	0001	$3+\sqrt{2}$
	8)	$00017 = 00000$	5

Inspection of the two sets of path reductions shows that $A(4)$ matches $B(6)$, indicating the presence there of a nodal intersection. Further, $A(6)$ and $B(7)$ differ by only one digit and the same is true for $A(7)$ and $B(8)$. Simultaneously, the length of A increases from 4 to 5.414 and that of B increases from 4.414 to 5, making A now longer than B. Hence the curves intersect nonnodally within the grid square defined by the end points of the four sequences $A(6)$, $A(7)$, $B(7)$, and $B(8)$.

Determination of Area

A fairly simple procedure exists for determining the area enclosed by an arbitrary curve. Consider the curve 10127 shown in Fig. 13(a). The area under this curve, *i.e.*, its integral with respect to x, can be obtained by adding the area covered by four vertical columns of unit width along the x axis:

$$2\tfrac{1}{2} + 3 + 3\tfrac{1}{2} + 4\tfrac{1}{2} = 13\tfrac{1}{2} \text{ square units.}$$

This suggests the following procedure for determining area. Define a modifier B to represent the instantaneous distance of the curve from the x axis. Thus in Fig. 13(a), the area A is initially zero and the modifier B is equal to "2." The initial slope of "1" (*i.e.*, 45°) causes the area to increase by the amount of the first vertical column to $2\frac{1}{2}$ and the modifier to change to 3. The next slope, "0," causes no change in the modifier since the curve is running parallel to the x axis, but the area is increased by the amount of the modifier, *i.e.*, to $5\frac{1}{2}$. It should be noted that the modifier gives the value of the leading edge of the particular vertical column. In the next step, the slope "1" increases the area by the modifier plus $\frac{1}{2}$ (to 9), and then increases the modifier to 4.

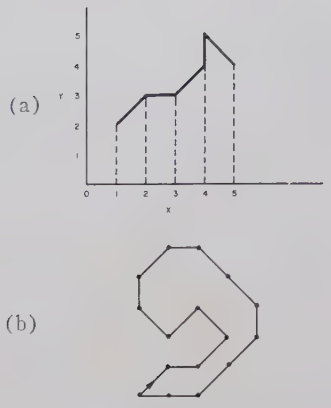


Fig. 13—Arbitrary curves to illustrate integration.

The slope "2" gives no change in area but increases the modifier to 5. The next slope, "7," increases the area by the modifier minus $\frac{1}{2}$ (to $13\frac{1}{2}$) and decreases the modifier to 4.

The foregoing procedure can be generalized to develop a set of rules for all eight slopes, as shown in Table IV. The table gives the slope (N), the change in area resulting from this slope (*i.e.*, the area in the vertical column), and the change to be made in the modifier. If the starting point coincides with the origin, the modifier is initially zero, if it does not, one must use the appropriate initial-condition value.

TABLE IV

Slope	Change in Area	Change in Modifier
0	$+B$	$+0$
1	$+B+\frac{1}{2}$	$+1$
2	$+0$	$+1$
3	$-B-\frac{1}{2}$	$+1$
4	$-B$	$+0$
5	$-B+\frac{1}{2}$	-1
6	$+0$	-1
7	$+B-\frac{1}{2}$	-1

As an example consider the curve 1013532107765544, shown in Fig. 13(b). This curve is traversed in a positive (clockwise) sense and has an area of 11 square units. The initial value of B is taken as zero.

N	A	B
1	$\frac{1}{2}$	1
0	$1\frac{1}{2}$	1
1	3	2
3	$\frac{1}{2}$	3
5	-2	2
3	$-4\frac{1}{2}$	3
2	$-4\frac{1}{2}$	4
1	0	5
0	5	5
7	$9\frac{1}{2}$	4
7	13	3
6	13	2
5	$11\frac{1}{2}$	1
5	11	0
4	11	0
4	11	0

Table IV can be expressed in the form of an algorithm and programmed for a digital computer. It is then a simple matter to determine the area enclosed by an arbitrary curve encoded according to the three-bit rectangular array. For open curves, single-valued in x, one can readily obtain their integral with respect to x. Integration with respect to y can be accomplished by first rotating the curve 90°.

Conclusion

A procedure has been developed for the description and encoding of arbitrary geometric curves which will facilitate the computer manipulation of such curves. Various approaches to the twin problems of describing and encoding have been discussed. Particular attention has been given to the rectangular-array type of representation which is one of the simplest and most natural, and one which lends itself easily to manipulation. Elementary manipulative techniques for the rectangular array have been described. Although consideration has been limited to plane (two-dimensional) curves, extension to three or more dimensions is readily visualized.

An Accurate Analog Multiplier and Divider*

E. KETTEL† AND W. SCHNEIDER†

Summary—**In the time-division multiplier the product $x_1 \cdot x_2$ is formed by pulse-duration modulation with x_1 and amplitude modulation with x_2. The circuit can be arranged in such a manner that division by means of a quantity x_3 can be carried out simultaneously, the output being $x_1 \cdot x_2/x_3$. When transistor switches are employed the error is $1 \cdot 10^{-4}$ machine units only. The zero error for x_2, used for amplitude modulation, can be reduced to $2 \cdot 10^{-5}$ machine units.**

Introduction

IN ACCURATE electronic analog computers the linear computing elements may have a relative error of 10^{-4}. Comparably good electronic multipliers first became known through the papers of Goldberg and others.[1-3] The time-division multiplier described by these authors reaches this error limit and thus permits the accurate solution of many nonlinear problems. On the other hand, this multiplier has shortcomings that make its improvement desirable.

One problem concerns division. As a rule division is effected by reversing the multiplication in accordance with Fig. 1, *i.e.*, feedback of a high-gain amplifier by means of a multiplier. Since all time-division multipliers contain a low-pass filter, the feedback loop of this circuit must be stabilized: an additive capacitive feedback employing a condenser C connected in parallel to the multiplier is used for this purpose. By this means the bandwidth is much narrower in division than in muliplication and, moreover, is dependent on the magnitude of the divisor. Since the bandwidth of the time-division multiplier is narrow in any case, the divider is thus too slow.

When an error of 10^{-4} is stated in respect of multipliers, the error refers to the machine unit and not the product. The zero drift can then be just as high and is thus much higher than the drift of a chopper-stabilized operational amplifier. In Stieltjes integrals,

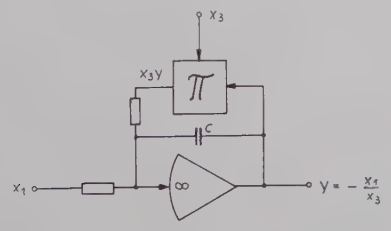


Fig. 1—Division by inversion of a multiplier.

an unpleasant phenomenon is often encountered in that the drift of the integrator can be worsened by two decades due to the preceding multiplier. The servo-multiplier does not have this error and has a zero drift not exceeding that of an inverter amplifier for one of its two inputs. For the corresponding input of a time-division multiplier the causes of the zero drift may be found solely in the errors of the electronic switch. More precise switches can be designed with transistors than tubes and the zero drift in the time-division multiplier can thus be improved considerably.

The Multiplier Circuit

Fig. 2 shows the well-known circuit of the time-division multiplier. The two multiplicands are represented by x_1 and x_2, b is a constant, y_1 is a commutator function and y_2 is the multiplier output. They are all nondimensional quantities in the range -1 to $+1$, referred to the machine unit. A and B are electronic switches assumed ideal. The upper part of the circuit consists of a loop with an integrator, a bistable flip-flop FF, and a switch A. The system is capable of oscillation and generates the commutator function y_1 which has the value $+1$ in each period for the duration T_1, and the value -1 for the duration T_2. The frequency is $F=1/(T_1+T_2)$ and the mean value $\bar{y}_1=-\alpha x_1/b$. The upper section is thus a pulse duration modulator controlled by x_1. The lower section of Fig. 2 then forms x_2y_1 by means of the switch B controlled by y_1, *i.e.*, modulates the amplitude of the commutator function. Behind the low-pass filter the mean value is $x_2\bar{y}_1$ and after amplification by factor $-1/\alpha$ provides the output $y_2=x_1x_2/b$. The pulse-duration modulator is an ideal relaxation oscillator. Its oscillation can be represented in a plane with the input z and output y_1 of the flip-flop as coordinates. The hysteresis characteristic Fig. 3 is then obtained, the time passing round the characteristic in an anticlockwise direction. The flip-flop break points are represented by $\pm a$. T_1, T_2 follow therefrom as the times the integrator requires to alter its output by $2a$, the sum of its two inputs being αx_1+b for T_1 and αx_1-b for T_2. It follows that

$$T_1 = \frac{2a\tau}{b + \alpha x_1}, \qquad T_2 = \frac{2a\tau}{b - \alpha x_1}. \tag{1}$$

Frequency and mean value of y_1 are then

$$F = \frac{b(1 - \bar{y}_1^2)}{4a\tau}, \qquad \bar{y}_1 = -\alpha x_1/b. \tag{2}$$

The frequency is proportional to the input b of the modulator switch. Large frequency variations are most

* Received by the PGEC, July 21, 1960.
† Telefunken Research Institute, Ulm, Germany.
[1] E. A. Goldberg, "A high accuracy time division multiplier," *RCA Rev.*, vol. 13, pp. 265–274; September, 1952.
[2] D. C. Morrill and R. V. Baum, "Stabilized time-division multiplier," *Electronics*, vol. 25, pp. 139–141; December, 1952.
[3] S. Sternberg, "An accurate electronic multiplier," *RCA Rev.*, vol. 16, pp. 618–634; December, 1955.

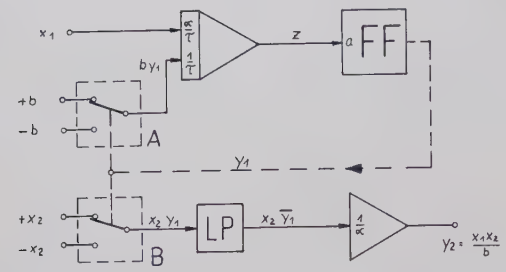


Fig. 2—The customary circuit of a time-division multiplier.

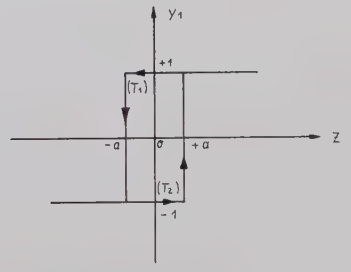


Fig. 3—Hysteresis function of the flip-flop-*FF* in Fig. 2.

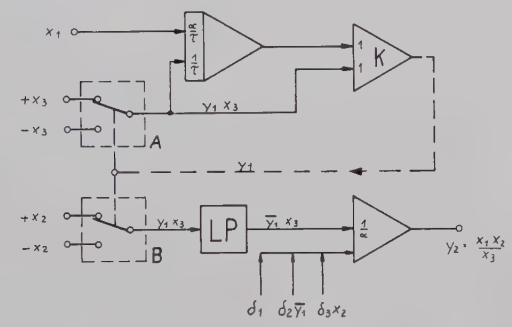


Fig. 4—The circuit for the time-division multiplier and divider

undesirable since the low-pass filter incorporated in the multiplier is a difficult compromise between multiplier bandwidth and attenuation of the frequency F, and large frequency variations would make the realization of the compromise even more difficult. Therefore, b must be constant and in general is equal to machine unit 1.

If a third, variable input quantity $x_3=b$ is applied to the modulator switch A, the frequency F is independent of x_3 only if b/a remains constant, and this demands the flip-flop breakpoints to be equal or proportional to x_3. The output $y_2=x_1x_2/x_3$ follows therefrom. Apart from multiplication the equipment also permits division. The hysteresis function (Fig. 3) is described mathematically by $y_1=\mathrm{sgn}(z+ay_1)$ and may be generated by positive feedback of a comparator with ay_1. A hysteresis function with breakpoints $\pm x_3$ is then obtained by means of feedback of a comparator K with the output x_3y_1 of the modulator switch A as shown in the circuit (Fig. 4.) Hence, now

$$F=\frac{1-\bar{y}_1^2}{4\tau},\qquad \bar{y}_1=-\alpha x_1/x_3,\qquad y_2=x_1x_2/x_3. \quad (3)$$

In the new circuit the pulse duration modulation is effected by means of quotient x_1/x_3. The modulator oscillates only if x_3 is greater than 0, a condition valid in principle for every divider. With consideration to the necessary limitation of the time ratio T_1/T_2, $|x_1|$ must be less than, or equal to, x_3. The ratio of minimum to maximum frequency is then $1-\alpha^2$ and is exactly as high as in the multiplier designed in accordance with the circuit (Fig. 2).

The Transistor Switches

For realizing the switch in Fig. 4, transistors are more suitable than tubes. In view of the I_{co} currents they have a poor blocking behavior but, on the other hand, have a very well-defined conductivity, particularly in the inverted connection where the functions of emitter and collector are interchanged. In Fig. 5 the residual voltage of an alloy Ge-transistor in inverted connection is shown. At very low emitter currents I_E, as must be supplied by the switch in Fig. 4, the residual voltage is less than 1 mv and, in addition, is only slightly influenced by temperature. Fig. 6 shows the temperature coefficient of the residual voltage for $I_E=0$ as function of the base current. Over a wide range of base current I_B, the temperature coefficient remains below 4 μv/°C. By employing transistors then, voltage switches can be designed that cause errors not higher than 10^{-4} even at a machine unit of 10 v. Fig. 7 shows the simplest type of transistor switch that can be utilized for the time-division multiplier. The transistors Tr1 and Tr2 are conductive alternately, the base currents I_B flowing via the collectors corresponding to the inverted connection. By means of the transistor conducting, the inverse current of the blocked transistor is shorted and thus does not influence the output current I_2, provided that the internal resistance of the voltage source U_1 is small. Since, in contrast to tubes, transistors can conduct currents of different polarity (Fig. 5), the switch functions independently of the polarity of U_1. The switch in Fig. 7 was employed by Schmid,[4] in a time-division multiplier attaining a product error of 0.1 per cent. The error is higher than might be expected from the residual voltages of the transistors and is attributable to the time errors of the switch. In the switch shown in Fig. 7 the base current I_B and the required control voltage of the transistors are dependent to a great extent on the input voltage U_1 so that the storage time also changes with U_1. To design an accurate switch the base current must be kept constant. A further problem in respect to this switch is the loading of the voltage source U_1 by the base current, particularly by the peak base current, and it is not certain whether a preceding operation amplifier can handle this load. In our multiplier these difficulties were mastered by means of the switch Fig. 8. The switch is

[4] H. Schmid, "A transistorized four-quadrant time-division multiplier with an accuracy of 0.1 per cent," IRE Trans. on Electronic Computers, vol. EC-7, pp. 41–47; March, 1958.

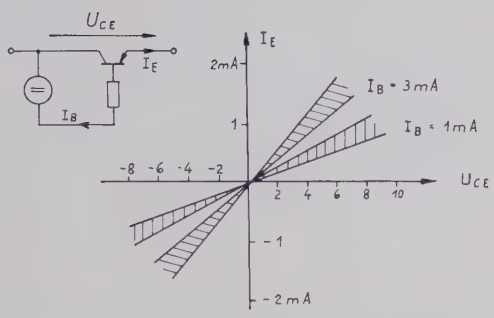


Fig. 5—Residual voltage characteristics of a transistor OC 613 in inverted connection. U_{CE} in mv.

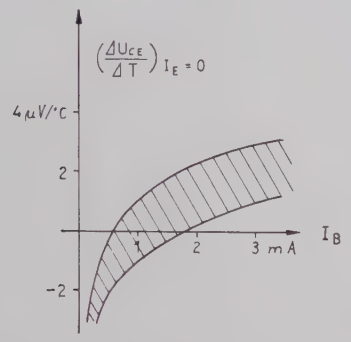


Fig. 6—Temperature coefficient of residual voltage in transistor OC 613 when emitter current I_E is zero.

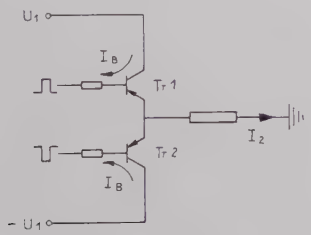


Fig. 7—Simple voltage switch with transistors in inverted connection.

completely balanced and the transistors are controlled via transformers. It contains four transistors in inverted connection and the pairs Tr1, Tr2 and Tr3, Tr4 conduct alternately. The control voltage required between base and collector amounts to only a few tenths of a volt and is independent of U_1. Silicon diodes D_1 and D_2 are used to limit the stationary base current to a constant value. The peak base current, important for the storage and rise times of the switch, is supplied from the transformer output via capacitances, different current paths being utilized for switching on and off in order to avoid undesired transient responses due to stray inductances. It has proved useful to set, by means of resistance R_1 and R_2, etc., such peak base currents that the transistors have the same storage time. In this manner the time difference of switches A and B in Fig. 4 is reduced to some 10^{-9} seconds. The advantage of the switch in Fig. 8 is also that the base current does not load the input terminals. Low, capacitive currents against ground and pulse currents due to short overlappings in time of the two switches can be shorted by capacities C_0.

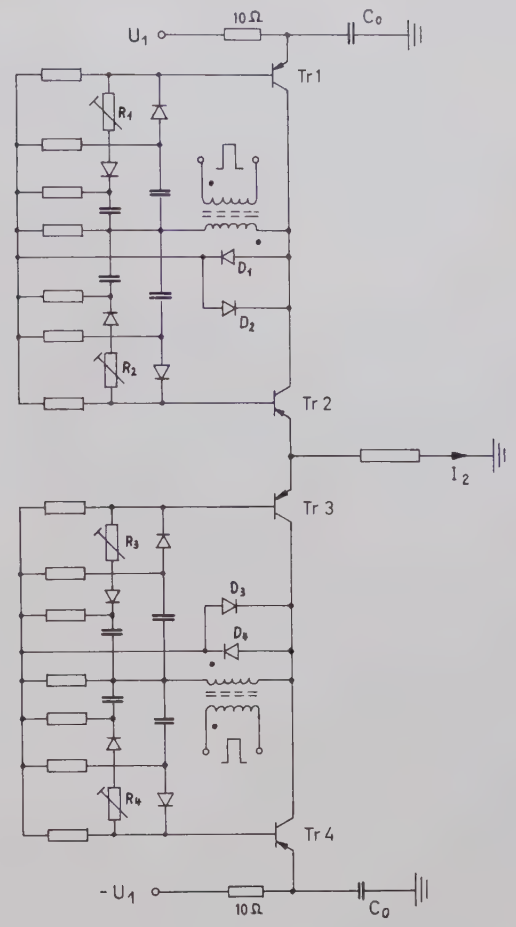


Fig. 8—The transistor switch employed in the multiplier/divider.

Results

The errors of electronic multipliers are generally referred to the maximum output range, which is equal to the double machine unit (MU). However, it appears to us more appropriate to indicate the error in machine units. Therefore, our values must be halved when comparing them with error data given by other authors.

The static errors of the multiplier described can be attributed almost exclusively to the errors of the electronic switch. The origins of the errors are the residual voltage of the transistors and different delay times of switches A and B caused by the storage effect. At a frequency of 10^4 cps the errors of our switches are 10^{-4} MU approximately. However, the zero error can be reduced considerably by three compensating currents which in Fig. 4 are applied to the summing junction of the output amplifier. The first current is derived from the machine unit and corrects the zero error at $x_1=0$, $x_2=0$, and the second is proportional to $\bar{y}_1$ and corrects the error at $x_1\neq 0$, $x_2=0$. The third current is proportional to x_2 and compensates the error at $x_1=0$, $x_2\neq 0$.

In Fig. 9 the zero error of the multiplier achieved in this manner is shown. The error is very small, particularly for $x_2=0$, *i.e.*, the quantity with which amplitude modulation is effected. After correction the error is less than $1\cdot 10^{-5}$ MU for any value of x_1. A change in tem-

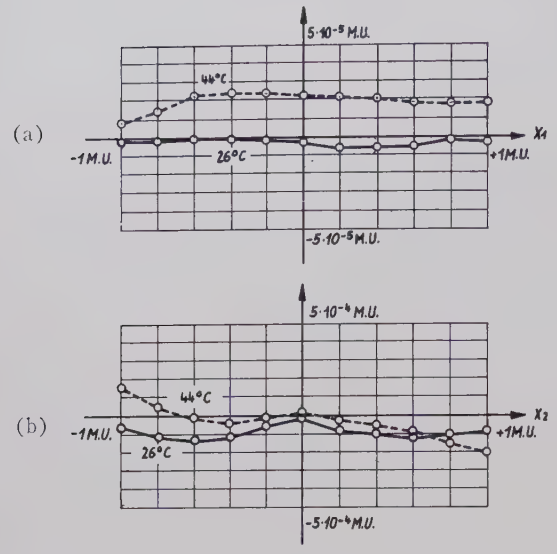


Fig. 9—Zero error of the time-division multiplier. (a) -1 MU $\leqq x_1 \leqq +1$ MU, $x_2=0$, $x_3=1$ MU. (b) $x_1=0$, -1 MU $\leqq x_2 \leqq +1$ MU, $x_3=1$ MU.

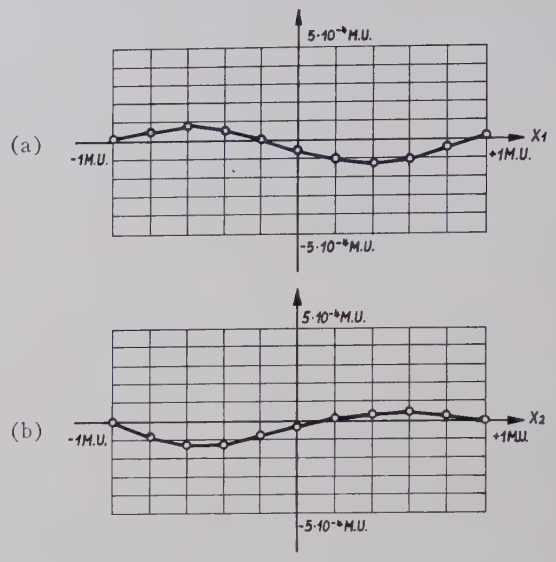


Fig. 10—The error of x_1x_2/x_3. (a) $x_2=1$ MU, -1 MU $\leqq x_1 \leqq +1$ MU, $x_3=1$ MU. (b) -1 MU $\leqq x_2 \leqq +1$ MU, $x_1=1$ MU, $x_3=1$ MU.

perature of 20°C changes the error by $2\cdot10^{-5}$ and the drift observed during one week was of the same magnitude. On an average the zero error for $x_1=0$ is $1\cdot10^{-4}$. A temperature change of 20°C and one week's drift alter the error by a further $1\cdot10^{-4}$ at the most.

The product error at $x_1=1$ or $x_2=1$ can be seen from Fig. 10. On an average the error is not more than $1\cdot10^{-4}$ and its maximum remains less than $1.5\cdot10^{-4}$. Even the changes within one week and at 20°C temperature change do not exceed this limit.

Due to the comparator delay time, the modulator frequency is not independent of x_3 at very low values of x_3, as would have been expected in theory. In practice, the divider has a sufficiently low frequency deviation for $x_3 \geqq 10^{-2}$ MU. The error of the quotient x_1/x_3 must increase when x_3 decreases because it is equal to the residual voltage of the modulator switch A referred to x_3. Fig. 11 shows the measurement of the error for the quotient $x_7/x_3=1$. The quotient of two quantities equal to 10^{-2} MU can be calculated with an error of less than 1 per cent.

The frequency of the modulator is 10^4 cps at $\bar{y}_1=0$. Insofar as the errors are attributable to the different switch delay times they are proportional to the frequency. On the other hand, the higher the frequency, the wider the bandwidth; and the dynamical errors decrease. Frequency selection is therefore a compromise between statical and dynamical errors. By dynamical errors expressed in MU, we imply the amplitude of the error vector of a frequency ω at the multiplier output, independent of whether it is an amplitude or phase error. Fig. 12 shows the dynamical error for a low-pass filter that attenuates the modulator frequency by 70 db. The error is an amplitude error because the phase is compensated for low frequencies. When ω is 25, it is $1\cdot10^{-4}$ and increases quadratically with the frequency.

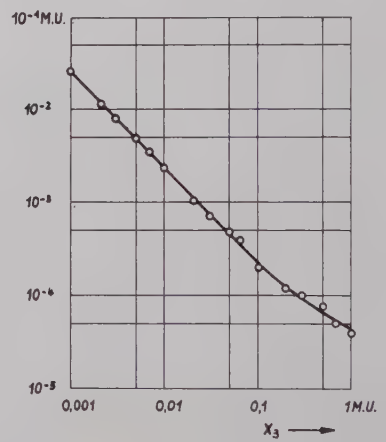


Fig. 11—The error of the quotient x_1/x_3 for $x_1=x_3$.

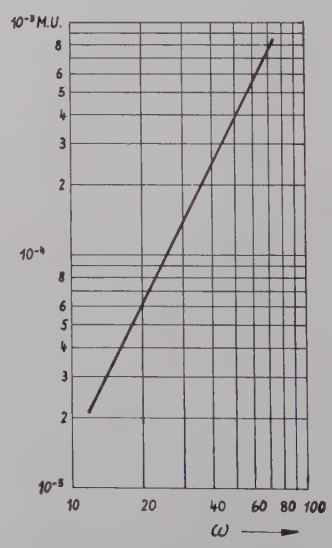


Fig. 12—The dynamical error of the time division multiplier for the output frequency ω.

High-Speed Analog-to-Digital Converters Utilizing Tunnel Diodes*

R. A. KAENEL†, MEMBER, IRE

Summary—Two analog-to-digital sequential converters have been devised which combine in one tunnel-diode pair per bit the functions of an amplitude discriminator and memory. In addition, one of the two schemes utilizes each tunnel-diode pair as a delay network. The conversion duration of one of these six-bit converters, which employs germanium 2N559 transistors and gallium arsenide 1N651 tunnel diodes, has been set to 1 μsec. Shorter conversion times are possible, but are not required in the present application of that converter as an integral part of an electronic high-speed signal processing system. The use of tunnel diodes presents a significant improvement in the art of converter design by virtue of circuit simplicity and performance.

The paper describes the principle and operation of the converters and discusses pertinent considerations for their design. Particular emphasis is given to the discriminator property of a series-aiding tunnel-diode pair.

A comprehensive bibliography relating to tunnel-diode switching circuits is attached.

I. Sequential Conversion

SEQUENTIAL converters, specifically the kind described in this paper, have many advantages over other schemes. For one, the speed requirement on the various components is not as high as in other methods of conversion. Another important factor is that conventional multipurpose elements such as diodes and resistors can be used to implement the principle.

Over the past few years high-speed analog-to-digital converters have enjoyed a rapidly increasing demand. The need for this kind of circuit stems from recent widespread use of digital computers for data processing, and from the fact that voice and other communication networks with PCM are also beginning to make use of the reliability of binary systems.

The principle of sequential converters, often called digit-at-a-time converters, has been described by Smith,[1] but will be briefly repeated here for completeness. In $N+1$ conversion steps, a sampled signal-amplitude value I is encoded into one of 2^N binary numbers. First, the amplitude is compared with a standard reference amplitude representing the magnitude of the highest-order digit of the converter. If the standard amplitude is smaller than the amplitude sample, it is subtracted from the sample and the remainder is then compared with a standard amplitude representing the next highest-order digit. This process is continued until the lowest order or units digit is compared with the remainder of the sample and is found to be either larger or smaller. Each time the standard amplitude is smaller than the remainder of the sample with which it is compared, a binary 1 is recorded. Each time the standard reference amplitude is larger than the remainder, a binary 0 is recorded and the standard amplitude is not subtracted from the sample. Obviously the signal amplitude is encoded by a bisection technique.

Symbolically this conversion process can be described by the following algorithm, in which Δ_i = difference computed at the ith step (testing of the $(i-1)$th digit) of the sequence of one full conversion operation and k = constant of the reference amplitude sources:

1st step:

$$\Delta_1 = I - k \cdot 2^{N-1}.$$

2nd step:

$$\Delta_2 = I - kR_1 \cdot 2^{N-1} + 2^{N-2}.$$

where

$$R_1 = \begin{cases} 1, & \text{if } \Delta_1 > 0 \\ 0, & \text{if } \Delta_1 \leq 0. \end{cases}$$

3rd step:

$$\Delta_3 = I - kR_1 \cdot 2^{N-1} + R_2 \cdot 2^{N-2} + 2^{N-3},$$

where

$$R_2 = \begin{cases} 1, & \text{if } \Delta_2 > 0 \\ 0, & \text{if } \Delta_2 < 0. \end{cases}$$

ith step:

$$\Delta_i = I - kR_1 \cdot 2^{N-1} + R_2 \cdot 2^{N-2} + \cdots R_{i-1} \cdot 2^{N-i+1} + 2^{N-i},$$

where

$$R_{i-1}(\Delta_{i-1}) = \begin{cases} 1, & \text{if } \Delta_{i-1} > 0 \\ 0, & \text{if } \Delta_{i-1} \leq 0. \end{cases}$$

last step:

$$k \geq \Delta_{N+1} = I - k\,R_1 \cdot 2^{N-1} + \cdots R_N \cdot 2^0 > 0.$$

After one conversion is completed, the cycle repeats with the difference Δ_1, and new values of the step functions $R_i(\Delta_i)$ are memorized until the beginning of the subsequent conversion operation cycle. According to the well-known sampling theorem for frequency band-limited signals, the time sequence of the conversion

* Received by the PGEC, October 10, 1960; revised manuscript received, February 3, 1961.

† Bell Telephone Labs., Inc., Murray Hill, N. J.

[1] An excellent reference on encoders is: B. D. Smith, "Coding by feedback methods," PROC. IRE, vol. 41, pp. 1053–1058; August, 1953.

vectors $R(R_1,R_2,\ldots,R_N)$ is a representation of the sampled signal, provided the sampling rate equals at least twice the top signal frequency.

The negative feedback character of the conversion process should be observed. Evidently, the logic of the sequence of comparison operations successively diminishes the difference Δ_i. As in linear feedback systems, an inverter introducing a 180° phase reversal in the closed loop is necessary for negative feedback, so that the error signal Δ_i causes a change in the loop which in turn reduces the error.

II. Converter Building Block

Before entering into a detailed explanation of converters that utilize tunnel diodes, it is profitable to examine separately the tunnel-diode pair properties most pertinent to the contemplated application.

The three operations which are basic to the sequential conversion scheme are sequential timing for digit testing, amplitude discriminating on the Δ_i, and memorizing of the R_i. All three operations can be performed by a simple tunnel-diode pair. Later, one pair will be associated with each converter digit. Occasionally these diode pairs will be called digit pairs, especially when they must be distinguished from other tunnel-diode pairs.

Consider first the *timing operation*, in particular the use of tunnel-diode pairs in an active delay line. The composite static current-voltage characteristic shown in Fig. 1 suggests that a diode pair can be harnessed to respond to a trigger in much the same way as a monopulser circuit. To this end, imagine an inductor L as a load in series with the pair (Fig. 2), and a bias which stably holds one diode in its high-voltage state and the other in its low-voltage state. In response to a negative trigger across the diode pair, both these diodes will temporarily be forced to reside in their low-voltage state. The diodes then remain in this common state until the magnetic field in the inductor increases and either one of the two diodes fires to its high-voltage state[2] (see Appendix I). This returns the diode pair to its only stable state and terminates the switching cycle. Transition from the low-voltage positive characteristic portion to the high-voltage positive region occurs at a practically constant current I_p. Indeed, with tunnel diodes this transition is so rapid that during its occurrence the current change through the assumed inductor can be neglected. The difference current i minus the current given by the composite dc-characteristic charges up the inevitable spurious parallel capacitors so that the voltage across the diode pair rises accordingly.

The positive-going edge of the diode-pair pulse response can be used to trigger a subsequent inductively-loaded diode pair. In doing this, an active element such as a transistor would be introduced as a coupling link for

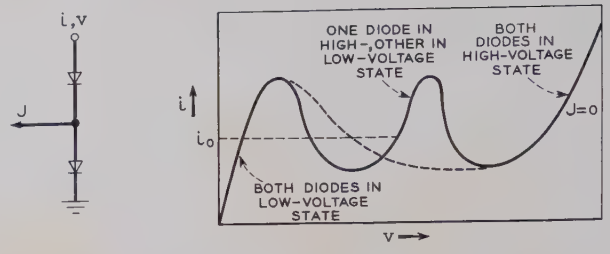


Fig. 1—Composite characteristic of a tunnel-diode pair.

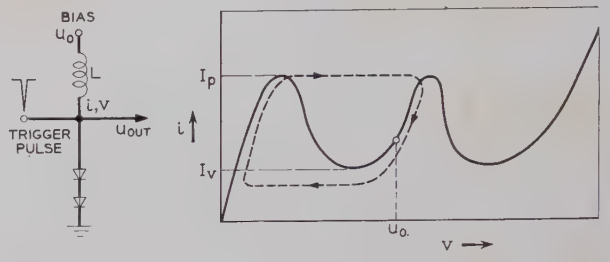


Fig. 2—Monopulser performance of a tunnel-diode pair.

most reliable results. This type of arrangement appears in one of the circuits described later. A chain of diode pairs coupled in this manner provides a timing circuit in which successive outputs deliver pulses which are delayed in time. With a quiescent load line that permits only one diode to reside in its high-voltage state, the other correspondingly in its low-voltage state, the diode-pair response to a trigger will always result in the same response voltage oscillograms.

The *discriminator property* is revealed by tracking the performance of the diode pair on the characteristic of each individual diode (Fig. 3). Assume that at time t_2 the diode-pair load line, which in the quiescent condition had intersected the center stable composite characteristic portion, has been shifted to intersect the low stable characteristic portion only. This has caused both diodes to occupy their low-voltage states. Let now the diode-pair load line move back toward its quiescent condition so that the pair current i begins to build up. Shortly before, or precisely at the point at which the common current i reaches the remarkably stable diode peak-current I_p (defined in Fig. 2), the net current through one of the two diodes first exceeds the value of I_p. The load line then no longer intersects the composite diode-pair characteristic in the stable low-voltage portion. If at that time the instantaneous load line intersects the center stable characteristic branch, between times t_3 and t_4 either the top or the bottom diode fires to its high-voltage state while the other diode remains in its low-voltage state. The decision as to which diode switches depends on the flow direction of the control current J (Fig. 1).

Once this decision has been made, a control current $|J|<(I_p-I_v)/2$ cannot affect the diode states reached as long as the quiescent load line intersects the composite characteristic at a current value $(I_p+I_v)/2$ (Fig. 4). Thus, with the control current fluctuations ac-

[2] R. A. Kaenel, "Novel adder-subtractor circuit," 1960 IRE WESCON Convention Record, pt. 3, pp. 53–64.

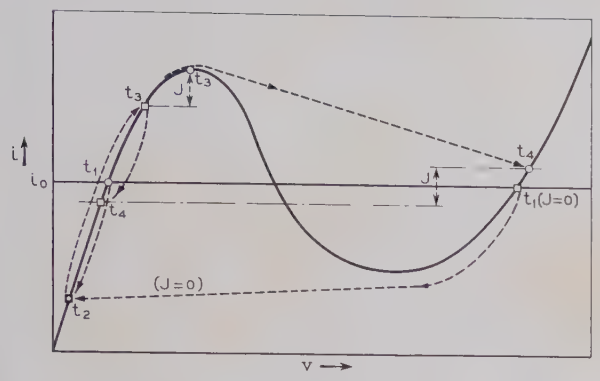


Fig. 3—Discriminator action of a tunnel-diode pair. As the current i through both diodes approaches the crest current I_p, the diode with the current head start due to J will first fire to its high-voltage state. With a proper load line, the other diode is thus captured in its low-voltage state.

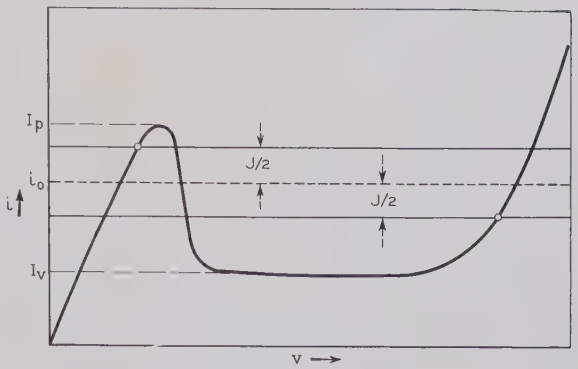


Fig. 4—Quiescent conditions of two diodes in a pair. Uncritically small currents J cannot affect the state of the two diodes.

cordingly limited and the quiescent current properly adjusted, a decision reached is reliably *memorized* by the states of the diodes of the diode pair.

The discrimination process for small control currents J can fail to result in a firm decision. Instead, both diodes can begin to switch towards their high-voltage states. This might be detrimental to the converter application of tunnel-diode pairs, and hence a little elaboration on this subject is warranted. After examining the onset of switching of the two diodes, the separation of their end states as it is required for proper discriminator performance will be discussed. Temporarily, the process of diode-state separation will be assumed to be caused by a load line that intersects only the center stable portion of the static diode-pair characteristic.

For two reasons the switching of both diodes to the high-voltage state must be banned in the converter application with which we are concerned. First, the trigger response of each converter-digit diode pair varies depending on whether one or both diodes switch, which might adversely affect the circuit timing performance. Secondly, and more important, the quiescent voltage across one diode can assume three stable values rather than only the two required by the binary conversion scheme, *viz.*, one low-voltage and two high-voltage values corresponding to both diodes residing in their low-voltage states, or their high-voltages states, or both diodes residing in different voltage states, one high and the other low. Usually the diode-pair current that flows when both diodes are in their high-voltage states differs from that flowing when the two diodes are in separate states. Accordingly, the high-state voltage can assume two values which are related to the two current values.

To delineate the possibility of both diodes switching towards their high-voltage state, consider a diode-pair load that consists of a resistor R in series with a voltage source, and set $J=0$ (Fig. 5). Assume both diodes to be in their low-voltage state and assume that the voltage u of the source across the series combination of diode-pair and resistor increases linearly from zero with a slope α. Let the voltage across the two diodes be v, the

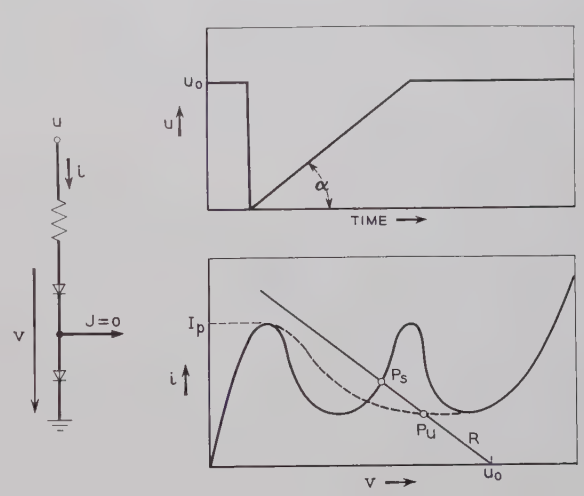


Fig. 5—Example for illustrating the possibility of joint switching of both pair diodes.

current through them i. Ohm's law for the loop requires that

$$u = i \cdot R + v.$$

Ideally, when the current i reaches I_p at least one diode begins to switch towards its high-voltage state. If the increase in v due to this switching action surpasses the increase in u, the voltage across the resistor R decreases and thus the current i will not overshoot I_p, and, in fact, will diminish. However, since the switching action starts out from a current crest that causes the initial switching-voltage slope to be zero, the current i will always exceed I_v (because of the further increase of the voltage difference $u-v$), and both diodes will switch provided they are sufficiently closely matched in I_p.

The problem of both diodes of a digit pair switching toward their high-voltage state can be eliminated altogether with the aid of a properly timed auxiliary tunnel-diode pair discriminator which controls the digit diode-pair current J (Fig. 6). This discriminator can be a common part of several digit diode pairs and furnishes to them (through a power amplifier stage) either a positive or negative control current J so that they now operate in a pure binary fashion with no marginal conditions ($J=0$) occurring. The auxiliary diode-pair

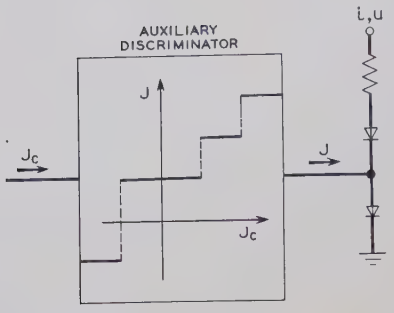


Fig. 6—Auxiliary circuit that prevents the diode-pair control current J to be zero at the instance of diode-pair discrimination action.

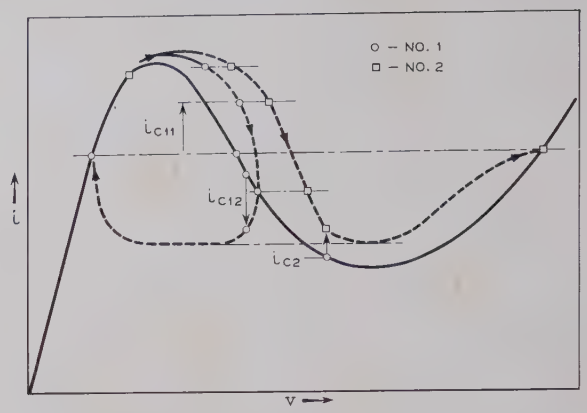


Fig. 7—Transient relaxation process when both diodes start to switch simultaneously. No. 1 designates the top diode of the digit pair and No. 2 the bottom one, i_c=current that charges up spurious diode parallel capacitance, affecting diode voltage to change according to $dv/dt=i_c/C$. i_c being positive (i_{c11}) charges up the spurious capacitance, whereas when i_c is negative (i_{c12}) this capacitance becomes discharged.

discriminator is allowed to assume any one of the three stable states mentioned before because the digit tunnel-diode pairs can be made to distinguish only between two voltage states, one high and the other low, of the bottom diode of that discriminator, and not the exact voltage across it.

Even if both diodes start to switch, they will not necessarily both end in their high-voltage states. Assume that a resistive load line intersects the composite diode-pair characteristic in the center stable portion only. As will be discussed next, the two diodes which both have begun to switch will then rapidly relax to their only stable quiescent condition P_s (Fig. 5), in which one exhibits a high-voltage state and the other a correspondingly low-voltage state. The intersection P_u of the composite characteristic portion that is represented by the dotted line with the resistive load line is a point of precarious equilibrium. Namely, if both diodes resided in their negative-characteristic region, the center node of the diode pair would exhibit a negative resistance. Since that node is connected to a high-resistance control current source, this state is inherently unstable. A relaxation transient process would immediately throw the two diodes to their stable branches and thus separate the diode states.

The transient relaxation process for a resistive load occurs in the following sequence of events. Initially let both diodes start to switch, and if they fire simultaneously, switching occurs according to the dotted composite characteristic portion. As both diodes spurt towards their high-voltage states (refer to Fig. 7), the series current i decreases to follow the transient load line which is partially given by the load resistor and partially by the spurious inductors. Unbalances in the inherent diode-switching time constant (or a nonzero control current J) inevitably cause one diode to race the other diode with respect to voltage increase. Eventually the faster diode, say diode no. 2, can cause the current i to decrease to a value equal to the current value that is given by the dc-diode characteristic at the voltage of the slower diode (no. 1) at that time. Thus the difference current i_c which charges up the spurious capacitance of the slower diode becomes zero and the voltage across this diode ceases to increase. Further decrease in i then discharges this spurious capacitance causing the slower diode to switch back to its low-voltage state while the faster diode assumes its high-voltage counterpart. If the slower diode at the beginning of the retrograding process resides in a highly negative-resistance region, then it will reach its low-voltage state before the other diode has noticeably left the voltage state that existed at the outset. The final adjustment to the quiescent condition relaxes very rapidly. Evidently, utilizing perfectly matched diodes, small currents J are discriminated at the negative-characteristic portion rather than the current crest. The *discrimination sensitivity* is therefore limited by the degree with which the two diodes can be matched electrically over their entire characteristic, rather than in the current crest region only. However, with perfectly matched diodes it is conceivable that for $J=0$ the diode pair will not reach its quiescent decision state but, instead, will break into oscillation caused by simultaneous switching of both diodes. This is not a real default because in the intended application of the diode pair testing of the next-lower-order digit will always change the control current J such that this oscillation will be interrupted.

A clamping diode in parallel with the diode pair can modify a rather high-resistive load to intersect the composite characteristic reliably in one stable point, as suggested by Fig. 8. With an inductive load a clamping diode reduces the relaxation time ordinarily required for the two diodes to assume their quiescent condition. To be specific and as illustration, let both diodes fire to their high-voltage states. As they reach the clamping voltage the clamping diode starts to conduct, not instantaneously, but exponentially with the small, yet not negligible, time constant given by the spurious inductances. In the reversing process, one of the two diode operating points will cross the negative portion of the static diode characteristic before the operating point of the other diode does. This will return that diode to its low-voltage state, all under the control of the small spurious inductances. During the short time of

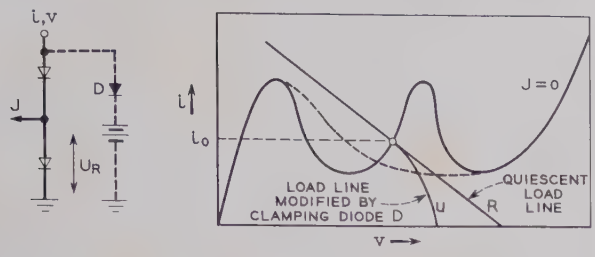


Fig. 8—Composite characteristic of two series-aiding tunnel diodes. The load line of a resistor has been added to illustrate the effect of a clamping diode *D*.

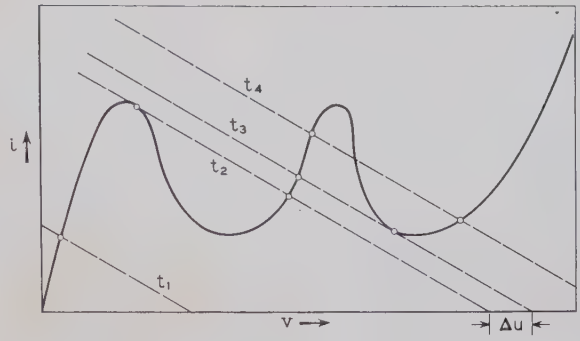


Fig. 9—Load line that, on its shift back to the quiescent state, temporarily intersects the center stable characteristic branch alone.

this sequence the current through the main inductor decreases slightly. This captures one diode in its low-voltage state. Without the clamping diode a similar order of events occurs. However, it takes much longer for the diodes to begin the retrograding process, because the timing is now determined by the rather large main-load inductor.

As a load line shifts from where it intersects the low stable branch of the static diode-pair characteristic to its quiescent position in which it intersects the center and high stable branch, both diodes can start to switch simultaneously to their high-voltage state without implying that they both will stably assume this state. To substantiate this situation, picture a resistive load line that intersects the composite diode-pair characteristic only in the center stable branch for those power supply voltages that lie in a narrow range Δu (Fig. 9). This kind of rather horizontal load line might be required to accurately control the quiescent diode current. Imagine now that both diodes start switching to their high-voltage state. Yet, because of the load line assumed, diode switching begins when the load line intersects only the center stable composite characteristic branch and, as a consequence, in the quiescent condition both diodes will not necessarily reside in their high-voltage state. Evidently, if the time interval t_i during which the load line intersects the composite characteristic in the center stable portion exclusively is long enough as compared with the relaxation time of the diode pair, then only one diode will assume the quiescent high-voltage state during the time t_i. This resulting configuration will then be preserved even if the quiescent load line intersects the composite characteristic in two stable points. The final quiescent diode configuration will be the one attained during the relaxation time interval t_i. Thus for only one diode to switch it is necessary that

$$t_i \cdot \alpha < \Delta u,$$

where, as before, α=slope of supply voltage increase (see Fig. 5).

In reality, the current J to the diode-pair tie point is not zero over an extended length of time; in particular, J is inherently disturbed with noise. If the J fluctuations are of a speed comparable with that of diode switching, as is usually the case, the dissymmetry caused by noise can improve the relaxation time. (This assertion can easily be verified by the reader himself.)

III. Complete Converter Configurations

A. Simplified Circuit

The converter, shown in simplified form in Fig. 10, results from utilizing the monopulser and discriminator properties of an inductively-loaded diode pair discussed above. The tunnel-diode pairs are so biased that in the quiescent condition only one diode resides in its high-voltage state, the other in its low-voltage state. To trace the circuit operation, assume that all bottom diodes are in their high-voltage states, designated 0, and assume that the signal input causes a current J at the tunnel-diode-pair tie point to flow outward. Of course, at the beginning of a conversion cycle, with all reference voltage sources (u_R) in their high states 0, the signal input will always cause a control current in that direction. The conversion trigger throws the bottom diode of the highest-order digit stage, which is connected to the weighting resistor R, to its low-voltage state, designated 1. Hence the current into the inverter decreases by a well-defined amount, say ΔJ. The current through the inductor now increases gradually, timed by the inductance, and approaches in magnitude the diode peak current. When the inductively-loaded diode pair regenerates, and if the current to the diode-pair tie point continues to flow outward, the bottom diode remains in its low-voltage state, and the top diode fires to its high-voltage state. Alternatively, if the control current changes the direction of flow because ΔJ is bigger than the original current contribution J caused by the signal, the bottom diode switches back to its 0 state and the original control current will flow again. At the moment when either the top or the bottom diode switches, the differentiated and inverted trailing edge of the diode-pair pulse response triggers the second converter stage connected to the weighting resistor $2R$, which tests the next highest-order digit. Accordingly the other digits are tested to complete one conversion cycle.

Interestingly enough, a control current that tends to reset to their high-voltages states the bottom diodes of

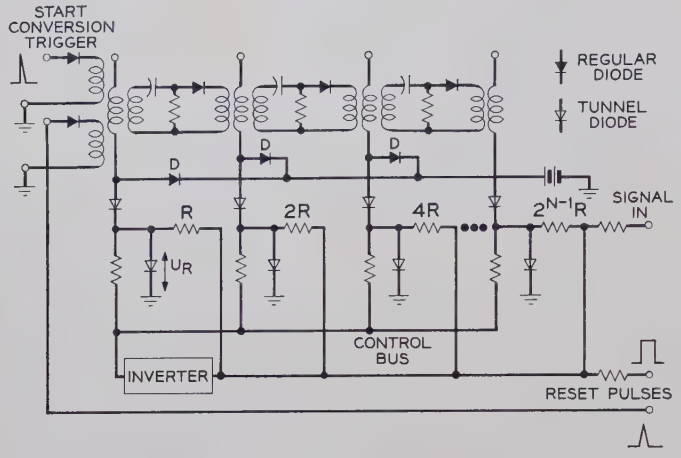


Fig. 10—Basic sequential analog-to-digital converter that utilizes an inductive tunnel-diode-pair load to achieve monopulser timing action. The trailing pulse edge of one stage is coupled to the subsequent stage to initiate digit testing.

the digit pairs, which assume the function of reference sources, is caused by a high voltage at the control bus. Correspondingly, a low voltage at the control bus tends to set these reference sources to their low-voltage state. For proper converter operation the control-bus voltage must be proportional to the inverted difference between the sum of the weighted converter-reference currents and the signal current processed. Obviously this is a feature typical of negative-feedback systems.

As in single-loop negative-feedback systems, the inverter input, which is connected to the current-summing bus, tends to remain at constant voltage so that the loading of each stage also tends to remain constant. In addition, since the voltage at the control bus accordingly tends to be invariant, it is of no concern that the reference voltage u_R varies with J, increasing when more of current J flows into the diode-pair tie point. However, means (which, for clarity, have been completely omitted in Fig. 10) must be provided to insure that during the time of digit testing the reference voltage of the digit being tested does not vary with series current i. One should realize that removing the inverter and control bus would not damage the basic operational principle of the circuit, and the negative-feedback character would be preserved, for the inverter has been added solely for the sake of the subsequent discussion.

Before discussing the converter reset mechanism and then two practical converter implementations, a supplementary remark on the tunnel-diode pair timing action is desirable. The control current J biases the diode which will fire to its high-voltage state and thus causes the relaxation time of the monopulser action to be reduced. Consequently, if the error current $|J|$ is large, the digit decisions are made faster than when $|J|$ is small. This is not really a default; after all, a decision on small-error currents must insure that small-switching transients have decayed sufficiently so as not to cause an error. With $|J|$ large, small transients are less critical and a decision can safely be made before these transients have completely vanished.

After each complete conversion sequence, the diode-pair stages must be reset to their 0 state. One advantageous reset procedure, which avoids much additional circuitry, alternately converts a zero signal amplitude. With control current J flowing inward because of zero signal, as a result of a trigger pulse, all bottom diodes fire to their 0 state. When using this reset scheme, the conversion trigger sequence must have a repetition frequency of at least four times the top frequency of the signal to be encoded. In synchronism with every other trigger pulse, the inverter input must be clamped to ground.

B. First Practical Circuit

In the simplified circuit discussed, the voltages across the bottom tunnel diodes of the pairs are used directly as a reference source. The bottom diodes are sequentially switched to their low-voltage states to test the digits they represent. Whether or not to reset the digit source is decided by a difference process. Depending on the sum current on the control bus, one of the two diodes of the pair switches to its high-voltage state. Incidentally, a highly accurate test-voltage pattern would be required to discriminate precisely and then reliably store the decision reached if no difference discrimination technique inherent with a tunnel-diode pair were used. In contrast, a tunnel-diode-pair discriminator can generate its own test-voltage oscillogram with an inductive load.

Incorporating current gain in the inverter relaxes the uniformity requirements for the diode characteristics and augments the discrimination accuracy. A unilateral transistor circuit is best suited for this purpose. Since a transistor in large-signal operation can be made to switch faster than in small-signal mode, one transistor switch inverter is best associated with each diode pair. The weighting resistors are then connected to the transistor switch output, and current summation is accomplished in a noninverting low-input impedance common-base transistor stage.

With each transistor inverter in cutoff for the 1 state, the control current becomes fully effective for discrimination decisions and is not partially wasted in transistor base connections. Furthermore, as is absolutely necessary for proper circuit operation, the reference voltage representing state 1 remains constant at least up to the instant of decision. The composite diode-transistor characteristic depicted in Fig. 11 illustrates this situation. Indeed, the current crest segment of the composite characteristic is practically identical with that of the tunnel diode. Unlike the requirement for the 1 state, it is not essential to drive the transistor switch completely into saturation for the 0 state. This statement can easily be verified bearing in mind that with the transistors not in saturation, the digit reference voltages vary inversely with the error current J in

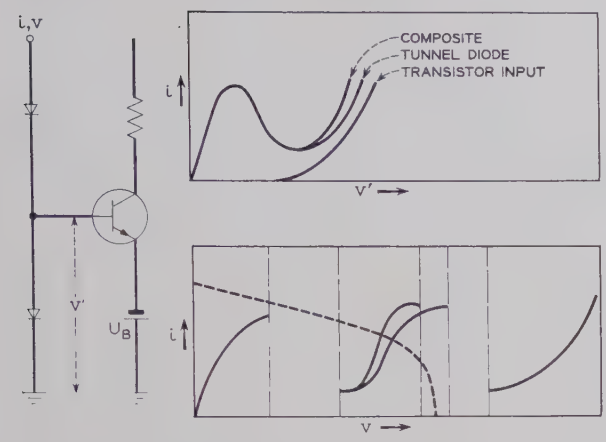


Fig. 11—Composite characteristic of two series-aiding tunnel diodes and one transistor. Because of transistor loading, the center stable characteristic portion splits into two branches.

the regular negative-feedback manner. Certainly, in the nonsaturating mode of operation, the storage delay, which insures that a digit is tested only after the decision for its preceding digit has been completed, can be incorporated in the common summing amplifier if the inherent transistor switch delay were to prove too small. With the timing function removed from the intrinsic converter (see Section III-C), the timing signal can be chosen accordingly to insure proper digit testing.

The static base-input capacitance of the transistor is small by comparison with the large spurious tunnel-diode capacitance (40 pf) and thus can be neglected. The Miller effect is initially not effective because of biasing of and response delay in the transistor. When the Miller effect magnifies the intrinsic base capacitance, tunnel-diode switching is decelerated. As discussed previously, if this retardation causes the tunnel-diode voltage to increase more slowly than the over-all voltage v, the diode current i can rise further and cause the second diode to switch also. To prevent this from occurring, it is necessary that a resistive load-line limit the current i to a value smaller than I_p for voltages at which the transistor switch starts or is conducting.

While the sluggish switching that stems from the Miller effect deteriorates discriminator performance, excessive minority carrier transit delay in the transistor can possibly annihilate this diode-pair discriminator property. Feeding the delayed transistor response back through the base-to-collector capacitor can reset the bottom tunnel diode and thus enable the top diode to assume its high-voltage state before the bottom one has had time to resume its original state. High peak-current diodes help to immunize against this effect. Other precautions include use of high-speed switching transistors whose frequency limitation is principally set by spurious capacitance, or insertion of a small inductor in series with the transistor base lead.

With an additional transistor stage, reliability can be enhanced by amplifying the trigger pulses that initiate digit testing.

To prevent the diodes from being switched by excessively large currents appearing at the diode tie point, in particular if induced by large signals, this control current J must be limited.

These considerations suggest the circuit shown in Fig. 12. Germanium tunnel diodes in the transistor emitters serve as voltage sources (0.5 volt) for biasing purposes in conjunction with gallium arsenide tunnel diodes and germanium transistors. They insure a strong transistor cutoff condition and speed up transistor switching.[3] Careful consideration must be given to the differentiating coupling transformers. Of course, the diode-pair trigger response could equally well have been differentiated at the coupling transistor output. In this case the emitter-to-base diode of the dc-connected trigger coupling transistor could substitute for the clamping diode D.

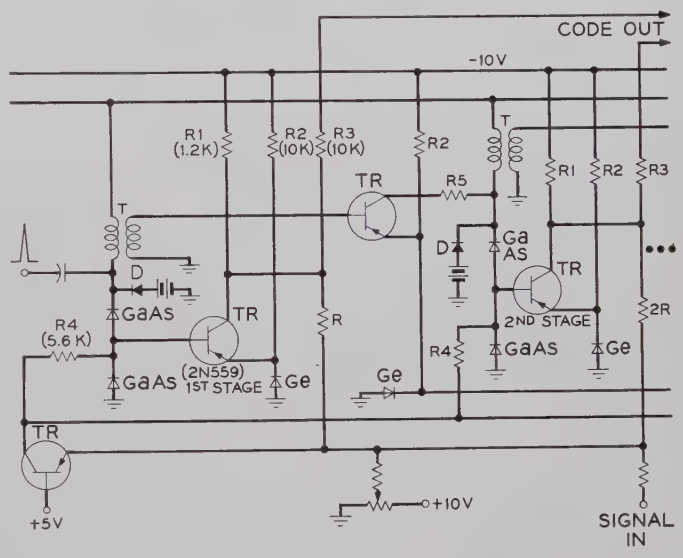


Fig. 12—Practical converter circuit that includes timing function in each digit tunnel-diode pair.

C. Second Practical Circuit

The circuit shown in Fig. 13 has the timing function removed from the diode pairs and assigned to a separate free-running shift register whose schematic has been redrawn and enlarged in Fig. 14 to improve clarity. Through *p-n-p* transistor switches designated by *TR*, the register content is coupled to the intrinsic encoder separately shown in Fig. 15. Of course, each diode pair could have been harnessed to perform also the shift-register functions. However, separating these functions from the intrinsic converter makes the design easier; the various time constants can then be considered and optimized separately.[4]

[3] The typically large junction capacitance of tunnel diodes is particularly helpful in this function.

[4] In another useful modification each diode pair would be biased bistably so that the pairs could reside in either the low- or center-voltage states. The clock pulse sequence would then consist of positive and negative spikes.

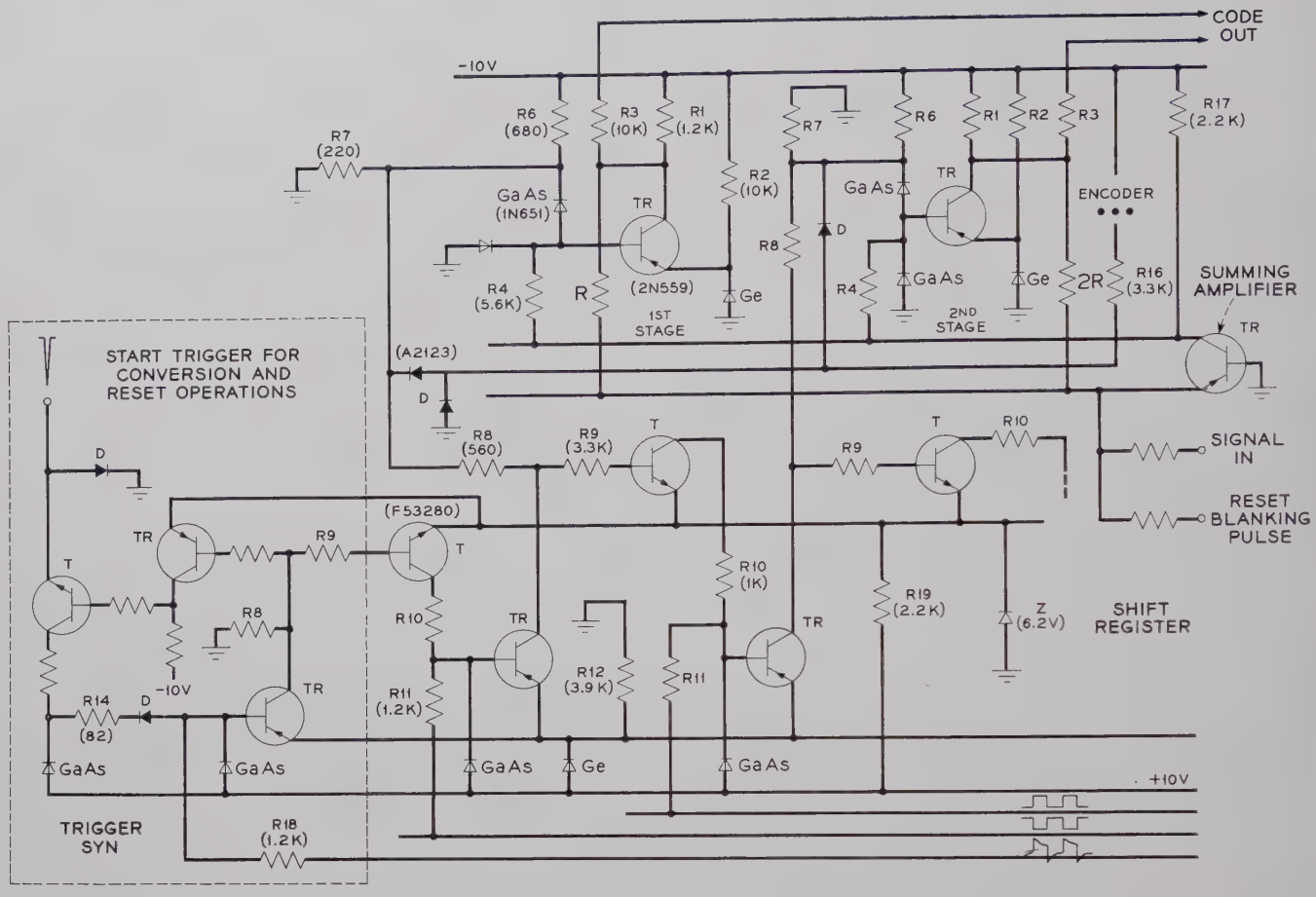


Fig. 13—Practical converter circuit that has timing function removed from each digit tunnel-diode pair.

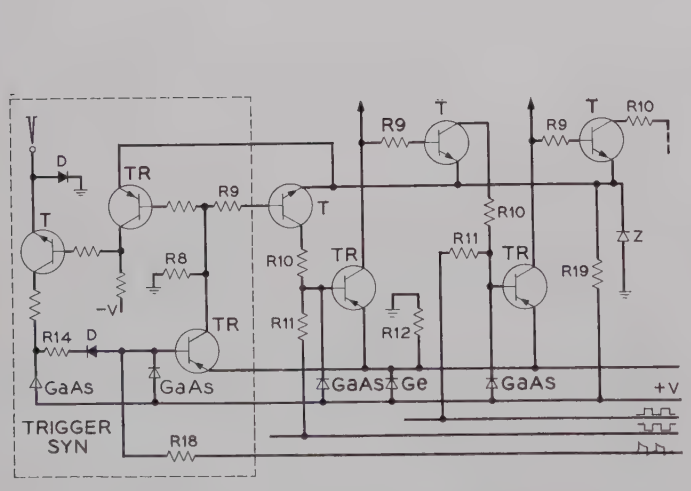


Fig. 14—Shift-register arrangements to generate a test-voltage pattern for the basic converter. The inherent delay in a *n-p-n* transistor *T* is sufficient to couple the content of one shift-register stage to its following stage. By means of the "Trigger Syn" circuit the conversion trigger is synchronized with the free-running clock shift pulses and then fed into the register.

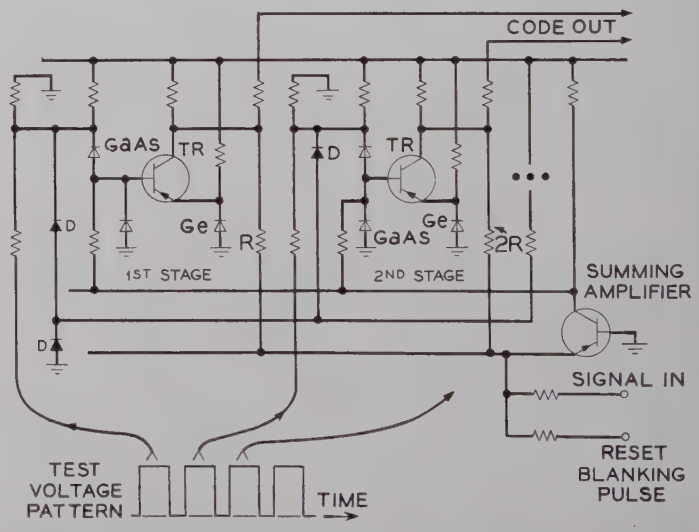


Fig. 15—Basic sequential analog-to-digital converter with externally generated test-voltage pattern for each tunnel-diode pair. Germanium tunnel diodes are used in the transistor emitters for biasing.

The intrinsic converter operates in a manner similar to that previously described. Initially, the test voltage source (see Fig. 15) causes the bottom diode of the first-stage diode pair to switch to the low-voltage state in which the top diode has been residing. The first-stage transistor thus becomes reverse biased. This in turn effects a current change through the weighting resistor R, and the current change, which corresponds to the weight of the most significant digit, combines with the signal current and causes the summing amplifier output voltage to increase. Depending on the magnitude of the current change relative to the signal current, the control current to the center nodes of the diode pairs reverses its flow direction and, at the outset of the second-digit stage testing, either one of the two diodes of the first-stage diode-pair switches to its high-voltage state to complete testing of the highest-order digit. Similarly, the lower-order digits are tested in proper sequence. After a completed signal-conversion sequence, as discussed previously, a zero signal is processed to reset the converter for the subsequent conversion sequence.

The shift register used to generate the test voltage for the converter is of rather standard design. The first two shift-register stages are necessary to synchronize the conversion trigger pulses, which initiate the conversion and reset sequences, with the register-shift pulses. A short trigger pulse, after having passed through the transistor gate of the trigger synchronization circuit, is memorized in the first GaAs tunnel diode. When this pulse has been shifted into the second stage of that synchronization circuit by the trailing edge of a shift pulse, the content of the first diode memory is erased. In proper sequence, the trigger pulse is then shifted into and along the inherent shift register, setting a register-stage diode while resetting its preceding stage diode. As will be explained next, the inherent delay of an *n-p-n* transistor (T), which is driven by a weak base current, is sufficient to couple the content of one register stage to its succeeding stage with the aid of a below-specified sequence of shift pulses.

The shift pulses fed to the trigger synchronization circuit assume three distinct amplitude levels, *viz.*, two negative and one positive (with respect to +10 volts). The most negative-amplitude level is reached by short spikes, each of which is marking one of the shift-pulse transitions from positive- to negative-amplitude levels. At this most negative level, the load line of the second GaAs diode intersects the electrical diode characteristic in both high- and low-voltage states when the first GaAs diode is in its low voltage state, and intersects it only in the high-voltage state when that diode is in its high state. Thus, assuming the first diode to have been switched to its high state, the second diode fires to that same state during the presence of the most negative shift-pulse level. The other negative shift-pulse level then holds the second diode in that high state while the transistor gate closes and, in so doing, resets the first diode to its low state. The second diode being in its high state now biases the following diode in the forward direction through the respective coupling transistor (T). However, this does not affect the state of this diode because the shift pulse amplitude for this diode is positive and of such magnitude that this diode can only reside in its low-voltage state. As the trigger synchronization circuit shift-pulse level becomes positive, the second diode resets to its low-voltage state and, in so doing, opens the transistor gate. At the same time the shift-pulse level for the third diode becomes negative. This negative-pulse voltage together with the *n-p-n*-coupling transistor current, which has been flowing because of the second diode being in the high-voltage state and is still partially flowing due to the inherent transistor response delay, cause the load line of that diode to intersect the characteristic in the high-voltage branch only and that diode switches to its high state. The negative shift-pulse voltage alone causes the load line to intersect the characteristic in both the low- and high-voltage regions, and thus reliably preserves the state that the diode has assumed at the beginning of the negative shift-pulse excursion. When the third diode is reset by the positive shift-pulse level, the state that diode has resided in is propagated to the subsequent diode of the inherent shift register and in proper sequence along the rest of this register.

To be precise, in this application the shift register acts like an electronic stepping switch.[5] Conventional shift register operation is possible if each information bit, which is to be stored in and then shifted along the register, is followed by a zero bit. Thus, two register stages are required to store one bit. Customary shift register performance could also be achieved if the circuit were driven by a one-phase shift-pulse sequence similar to that fed to the trigger synchronization circuit. Assume the one-phase shift-pulse amplitude to be at a negative voltage that permits each register diode to reside in either the high-voltage or low-voltage state. The following positive shift-pulse excursion then resets all of these diodes to their low-voltage states. Before the coupling transistors have responded to the resetting of the diodes, let the shift-pulse amplitude now become negative again, and in this process quickly reach a voltage level at which a diode remains in its low-voltage state except when it is biased by the coupling transistor. Consequently, a diode will switch to its high-voltage state if the diode preceding it resided in that high state; conversely, it will remain in the low-voltage state, if the diode preceding it resided in that low state.

[5] Instead of having only N distinct states like an ideal stepping switch, the shift register has $2^{N/2-1}$ states. N tunnel diodes connected in series can be forced by a particular common load line to perform like an ideal stepping switch with only one diode at a time permitted to reside in the high-voltage state. However, for active elements connected in parallel, such a load line exists only if the electrical characteristic of these elements is the dual of that of tunnel diodes.

This is obviously the action of a conventional shift register.

IV. Circuit Performance

The second of the described converters will presently be used in a high-speed electronic signal processing system that utilizes a magnetic core memory and tape transport. The sensitivity of each diode pair as a discriminator is better than 100 μa. No particular effort was exercised to improve this sensitivity, although it was found that 10-μa discrimination spread could readily be achieved. Of course, the tunnel-diode crest currents had to be matched by means of high-value resistors connected to the diode tie points and ± 10-volts power supply. The switching time of the the 2N559 transistors was estimated to be of the order of less than 20 nsec (1 nsc $= 10^{-9}$ sec).

The simplicity of circuit design is noteworthy. Indeed the converter configurations appear to be free of any requirement for exact component matching procedures, except for the values of the decoding resistors R. Components separately held to 10 per cent tolerances could generally be used.

Two converters in parallel double the signal amplitude conversion rate which is limited by the total processing time of one converter; namely, before an amplitude conversion of the first converter has been completed, the second one can begin its conversion on the subsequent amplitude sample. In particular, while one converter resets, the other can perform the conversion operation.[6]

Logical tunnel-diode pulse circuitry can be used to reduce the time duration needed for each amplitude conversion if this time is principally determined by transistor switching. Since this and similar kinds of speed-up techniques are not intended to be encompassed in this paper, the basic principle involved will only be outlined here. For example, the first two binary digits can be tested simultaneously by the following technique. As the first two reference sources representing the two most significant digits are switched to their 1 state, with the aid of three tunnel diode-pair discriminators, the hypotheses Δ_1 and Δ_2, each larger than zero, can be tested. Indeed, by checking the three threshold levels 0, $-k\cdot 2^{N-2}$, and $-k\cdot 2^{N-1}$, the differences Δ_1 and Δ_2 can be deduced:

$$\Delta_1 > 0 \quad \text{if} \quad \overline{D(-k\cdot 2^{N-2}) = 1}$$

$$\Delta_2 > 0 \quad \text{if} \quad \overline{D(-k\cdot 2^{N-2})\cdot D(-k\cdot 2^{N-1}) + D(-k\cdot 2^{N-2})\cdot D(0) = 1}$$

where

$$D(x) = 1$$

if level x is exceeded and

$$D(x) = 0$$

otherwise. Similarly the subsequent digit pairs can be tested simultaneously. This means of speed-up might be worthwhile applying for the following reason.

A delicate sample-and-hold circuit is unnecessary in a converter whose rate of the $N+1$ conversion steps is high relative to the top frequency of the signal to be encoded. In this case the signal amplitude remains practically constant during the conversion sequence. For voice frequencies, fast tunnel diodes enable a practically instantaneous conversion operation and thus obviate the need for such an additional refined circuit. This usually constitutes a considerable saving of circuit components and improves circuit reliability significantly.[7]

V. Conclusion

The converter design utilizing tunnel diodes has many advantages over a fully transistorized version. Besides the increased speed, the circuit has the virtue of simplicity. Nevertheless, one should refrain from systematically attempting to substitute high-speed tunnel diodes for all the transistors of the converter; tunnel diodes will never replace transistors, but they are complementing them efficaciously.

Appendix I

The monopulser action of a diode pair, as it is described qualitatively in the main body of the paper and is used for timing of the conversion sequence, can be deduced accurately from such action in a one-diode circuit (Fig. 16). After all, only one diode of the pair at a time is assumed to switch.

Initially let a trigger current at time t_0 switch the diode to its low-voltage state. When the very short trigger current is removed, the diode again draws the quiescent current i_0 which has kept flowing through the inductor L. The diode current then begins to increase exponentially. Combining a quadratic tunnel-diode characteristic approximation, which has its maximum at point (V_p, I_p) and goes through the origin (0, 0), with the differential equation for the circuit

$$L\cdot\frac{di}{dt} + V = U_0,$$

results in an equation of the form

$$L\cdot\frac{di}{dt} + k_1\cdot\sqrt{i} = k_2.$$

[6] Incidentally, the basic converter building block is exceptionally well suited for traveling-wave type encoders which inherently permit high conversion rates.

[7] It is worthwhile noting that, because of the timed sequential conversion process involved, no Gray-code type translator is required to keep level transition errors below a safe bound.

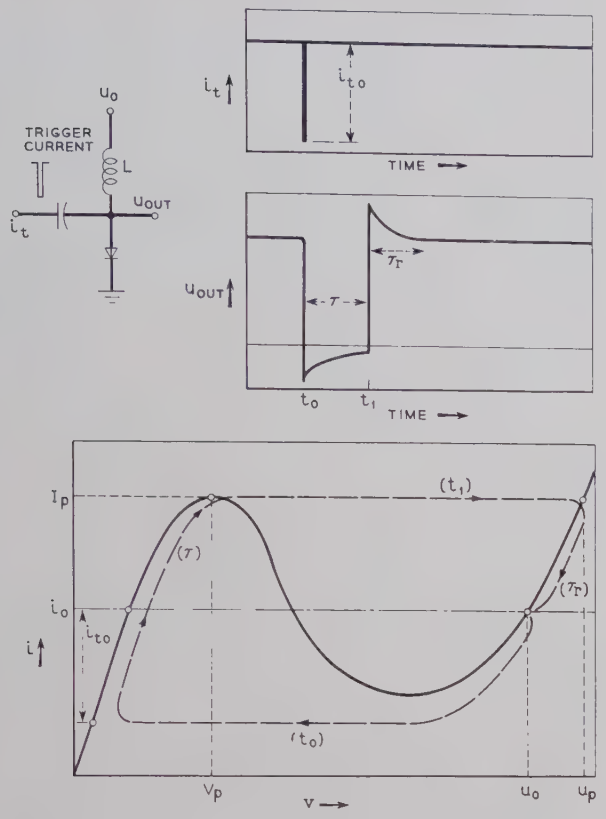


Fig. 16—Monopulser action of a tunnel diode.

This equation can be solved in closed form to give the time necessary for the diode current to build up to the value of I_p:

$$\tau = \frac{2I_P}{V_P} \cdot L \left\{ \sqrt{1 - \frac{i_0}{I_P}} + \frac{U_0 - V_P}{V_P} \cdot \log_e \frac{1}{1 + \frac{V_P}{U_0 + V_P}\sqrt{1 - \frac{i_0}{I_P}}} \right\}.$$

It is interesting to observe that the nonlinearity in the diode characteristic tends to linearize the voltage increase across the diode. Indeed, if $U_0 = V_p$ and $i_0 = 0$, then

$$\tau = \frac{2 \cdot I_P}{V_P} \cdot L.$$

When the diode current reaches I_p, the diode switches back to its high-voltage state (I_p, U_p). Exponentially, the diode current reaches its quiescent condition current i_0 which flowed at the outset of the diode monopulser cycle.

Appendix II

Proposed Circuit Symbol For Tunnel Diodes

A logical motivation exists for proposing a Θ (capital theta) as circuit symbol and a lower-case theta (ϑ or θ) as textual abbreviation for tunnel diodes. If a lower-case theta is properly fitted into a Cartesian coordinate system, the typical tunnel-diode characteristic is obtained (see Fig. 17). In addition, both *theta* and *tunnel diode* begin with the letter *t*. Furthermore, to be consistent with the nomenclature of regular diodes, the one proposed for tunnel diodes combines a regular polygon with a bar and indicates the forward biasing direction by an arrow (Fig. 17).

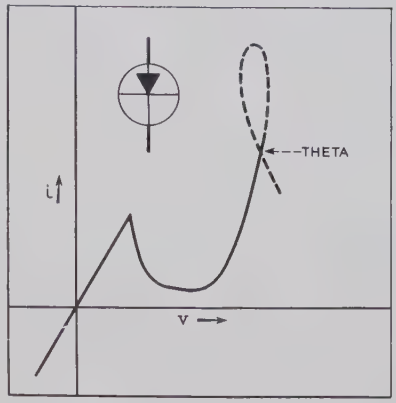


Fig. 17—Proposed circuit symbol for tunnel diodes.

The tunnel-diode symbol proposed elsewhere and reluctantly used in this paper can easily be confused with the symbol for a regular diode, especially if the printing is faded or vague. In contrast, the proposed symbol is distinctive, albeit similar. Even if almost any symbol is good once it has been generally accepted, a mnemonic symbol of the type proposed has merit.

Acknowledgment

The author gratefully acknowledges the effort of B. A. Stevens who compiled the attached bibliography. A search in some 50 internationally recognized publications laid the groundwork for this listing.

Bibliography on the Use of Tunnel Diodes as Switching Elements

This bibliography represents to the best of our knowledge the more significant publications which have appeared since 1959 to February, 1961.

[1] C. H. Alford, "Analysis and design of the twin-tunnel-diode logic circuit," 1960 IRE WESCON Convention Record, pt. 2, pp. 94–101.
[2] C. M. Barrack and M. C. Watkins, "Tunnel diode relaxation oscillators," *Electronic Design*, vol. 8, pp. 54–57; June, 1960.
[3] R. H. Bergman, "Tunnel diode logic circuits," IRE Trans. on Electronic Computers, vol. EC-9, pp. 430–438; December, 1960.
[4] R. H. Bergman and M. M. Kaufman, "Tunnel diodes in digital computers," *RCA Engr.*, vol. 6, pp. 14–17; August–September, 1960.
[5] W. F. Chow, "Tunnel Diode Digital Circuitry," presented at Internatl. Solid-State Circuits Conf., Philadelphia, Pa., February 10–12, 1960; *Digest of Tech. Papers*, Lewis Winner, New York, N. Y., pp. 32–33; 1960.
[6] W. F. Chow, "Tunnel diode logic circuits," *Electronics*, vol. 33, pp. 103–107; June, 1960.
[7] W. F. Chow, "Tunnel diode digital circuitry," IRE Trans. on Electronic Computers, vol. EC-9, pp. 295–301; September, 1960.
[8] W. F. Chow, U. S. Davidsohn, and Y. C. Hwang, *et al.*, "Tunnel Diode Circuit Aspects and Applications," presented at AIEE Winter Meeting, New York, N. Y.; January 31–February 5, 1961.
[9] L. G. Cox, "A tunnel diode crystal calibrator," *J. Brit. IRE*, vol. 20, pp. 621–623; August, 1960.
[10] U. S. Davidsohn, Y. C. Hwang, and G. B. Ober, "Designing

with tunnel diodes," *Electronic Design*, vol. 8, pp. 50–55, 66–71; February, 1960.
[11] E. M. Davis, "High Speed Esaki Switching Diodes," presented at Electron Devices Meeting, Washington, D. C.; October 27–28, 1960.
[12] L. Esaki, "New phenomenon in narrow germanium *p-n* junctions," *Phys. Rev.*, vol. 109, pp. 603–604; January 15, 1958.
[13] T. L. Francis and L. D. Wald, "Threshold Logic and Tunnel Diode Applications," AIEE Winter General Meeting, New York, N. Y., January 29–February 3, 1961.
[14] W. W. Gärtner, "Esaki or tunnel diode, part 2," *Semiconductor Products*, vol. 3, pp. 36–38; June, 1960.
[15] E. Goto, "Parametron and tunnel diode progress in Japan," Internatl. Solid-State Circuits Conf., Philadelphia, Pa., February 15–17, 1961, *Digest of Tech. Papers*, Lewis Winner, New York, N. Y., pp. 28–29; 1961.
[16] E. Goto, *et al.*, "Esaki diode high-speed logical circuits," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-9, pp. 25–29; March, 1960.
[17] E. Goto and the University of Tokyo VHS Computer Research Group, "On the Possibility of Building a Very High-Speed Computer with Esaki Diodes," Electronic Computer Tech. Committee, Tokyo IECEJ; October, 1959. (In Japanese.)
[18] H. K. Gummel and F. M. Smith, "Margin considerations for an Esaki diode OR gate," *Bell Sys. Tech. J.*, vol. 40, pp. 213–232; January, 1961.
[19] R. N. Hall, "Tunnel diodes," IRE TRANS. ON ELECTRONIC DEVICES, vol. ED-7, pp. 1–9; January, 1960.
[20] W. V. Harrison and R. S. Foots, "High Speed Switching Circuitry Using Tunnel Diodes," Internatl. Solid-State Circuits Conf., Philadelphia, Pa., February 15–17, 1961, *Digest of Tech. Papers*, Lewis Winner, New York, N. Y., pp. 76–77; 1961.
[21] Y. Hazoni, "Fast numerical analog converter," *Compt. Rend.*, vol. 251, pp. 343–345; July, 1960. (In French.)
[22] N. Holonyak and I. A. Lesk, "Gallium arsenide tunnel diodes," PROC. IRE, vol. 48, pp. 1405–1409; August, 1960.
[23] J. W. Horton and A. G. Anderson, "A full binary adder employing two negative resistance diodes," *IBM J. Res. & Dev.*, vol. 2, pp. 223–231; July, 1958.
[24] R. A. Kaenel, "Novel adder-subtractor circuit utilizing tunnel diodes," 1960 IRE WESCON CONVENTION RECORD, pt. 3, pp. 53–64.
[25] R. A. Kaenel, "One-tunnel-diode flip-flop," PROC. IRE, (Correspondence), vol. 49, p. 622; March, 1961.
[26] R. A. Kaenel, "Analog-Digital Converter Utilizing Tunnel Diodes," presented at Internatl. Solid-State Circuits Conf., Philadelphia, Pa.; February 15–17, 1961, *Digest of Tech. Papers*, Lewis Winner, New York, N. Y., pp. 100–101; 1961.
[27] M. M. Kaufmann, "Tunnel diode tenth microsecond memory," 1960 IRE INTERNATL. CONVENTION RECORD, vol. 8, pt. 2, pp. 114–123.
[28] H. R. Kaupp and D. R. Crosby, "Calculated waveforms for tunnel diode locked pair," PROC. IRE, vol. 49, pp. 146–154; January, 1961.
[29] G. Kesel, A. Ottmann, and H. N. Toussaint, "Germanium tunnel diodes for the high frequency range," *Nachrtech. Z*, vol. 13, pp. 191–195; April, 1960. (In German.)
[30] I. A. Lesk, N. Holonyak, and U. S. Davidsohn, "The tunnel diode—circuits and applications," *Electronics*, vol. 32, pp. 60–64; November, 1959.
[31] I. A. Lesk, N. Holonyak, U. S. Davidsohn, and M. W. Aarons, "Germanium and silicon tunnel diodes—design operation and application," 1959 IRE WESCON CONVENTION RECORD, pt. 3, pp. 9–31.
[32] I. A. Lesk and J. J. Suran, "Tunnel diode operation and application," *Elec. Engrg.*, vol. 79, pp. 270–277; April, 1960.
[33] I. A. Lesk and H. A. Jensen, "Germanium *P-N* Junction-Tunnel Junction Devices," presented at Conf. of the Professional Group on Electron Devices, Washington, D. C.; October 29–31, 1959.
[34] M. H. Lewin, "Negative resistance elements as digital computer components," *Proc. EJCC*, pp. 15–25, disc. pp. 25–27; December, 1959.
[35] M. H. Lewin, A. G. Sausenko, and A. W. Lo, "The tunnel diode as a logic element," Internatl. Solid-State Circuits Conf., Philadelphia, Pa., February 10–12, 1960; *Digest of Tech. Papers*, Lewis Winner, New York, N. Y., pp. 10–11; 1960.
[36] G. C. Messenger, W. Steiker, and C. D. Todd, "A survey of tunnel diodes," *Solid-State J.*, vol. 1, pp. 35–43; July–August, 1960.
[37] J. C. Miller, K. Li, and A. W. Lo, "The tunnel diode as a storage element," Internatl. Solid-state Circuits Conf., Philadelphia, Pa., February 10–12, 1960; *Digest of Tech. Papers*, Lewis Winner, New York, N. Y., pp. 52–53; 1960.
[38] G. W. Neff, S. A. Butler, and D. L. Critchlow, "Esaki diode logic circuits," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-9, pp. 422–429; December, 1960.
[39] G. W. Neff, S. A. Butler, and D. L. Critchlow, "Esaki (tunnel) diode logic circuits," International Solid-State Circuits Conf., Philadelphia, Pa., February 10–12, 1960; *Digest of Tech. Papers*, Lewis Winner, New York, N. Y., pp. 16–17; 1960.
[40] "Negative Resistance Diode Flip-Flop," in "Selected Semiconductor Circuit Handbook," S. Schwartz, Ed., John Wiley and Sons, Inc., New York, N. Y., pp. 6–45; 1960.
[41] L. S. Onyshkevych, W. F. Kosonocky, and A. W. Lo, "Parametric phase-locked oscillators—characteristics and applications to digital systems," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-8, pp. 277–286; September, 1959.
[42] J. A. Rajchman, "Solid-state microwave high-speed computers," *Proc. EJCC*, pp. 38–46, disc. pp. 46–47; December, 1959.
[43] J. A. Rajchman, "Computer memories. A survey of the state-of-the-art," PROC. IRE, vol. 49, pp. 104–127; January, 1961.
[44] G. N. Roberts, "Tunnel diodes," *Electronic Tech.*, vol. 37, pp. 217–222; June, 1960.
[45] M. Schuller and W. M. Gärtner, "Large Signal Circuit Theory for Negative-resistance Diodes in Particular Tunnel Diodes," presented at AIEE Winter General Meeting, New York, N. Y.; January 29–February 3, 1961.
[46] R. C. Sims, E. R. Beck, Jr., and V. C. Kamm, "A survey of tunnel-diode digital techniques," PROC. IRE, vol. 49, pp. 136–146; January, 1961.
[47] T. A. Smay and A. V. Pohm, "Thin Film-Tunnel Diode Logical Devices," presented at AIEE Winter General Meeting, New York, N. Y.; January 29–February 3, 1961.
[48] H. S. Sommers, Jr., "Tunnel diodes as high frequency devices," PROC. IRE, vol. 47, pp. 1201–1206; July, 1959.
[49] P. Spiegel, "High-speed scalers using tunnel diodes," *Rev. Sci. Instr.*, vol. 31, pp. 754–755; July, 1960.
[50] T. P. Sylvan and E. Gottlieb, "Tunnel diodes as amplifiers and switches," *Electronic Equipment Engrg.*, vol. 64; May, 1960.
[51] J. J. Tiemann, "Tunnel diodes and their use as multifunctional circuit elements," Internatl. Solid-State Circuits Conf., Philadelphia, Pa., February 10–12, 1960; *Digest of Tech. Papers*, Lewis Winner, New York, N. Y., pp. 8–9; 1960.
[52] C. D. Todd, "Combining transistors with tunnel diodes," *Electronics*, vol. 33, pp. 59–61; August 19, 1960.
[53] J. R. Turnbull, "100-Mc nonsynchronous computer circuitry using tunnel diodes," Internatl. Solid-State Circuits Conf., Philadelphia, Pa., February 15–17, 1961, *Digest of Tech. Papers*, Lewis Winner, New York, N. Y., pp. 74–75; 1961.
[54] J. W. Wentworth, "A description of the tunnel diode and its applications," IRE TRANS. ON BROADCASTING, vol. BC-6, pp. 21–30; September, 1960.
[55] H. S. Yourke, "Esaki Diode Inverter Logic Techniques," presented at Northeast Electronics Res. and Engrg. Meeting, Boston, Mass.; November 15–17, 1960.

Correspondence

Bibliography on Magnetostrictive Delay Lines*

Introduction

In recent years, there has been an increasing military and commercial use of magnetostrictive delay lines for pulse delay and storage in aircraft navigation, radar, radar target simulation, nuclear instrumentation, and the digital computer. This trend has been particularly evident in low-cost data processing and automatic control. The advantages of using this type of storage—namely, economy, temperature stability, high operating speed, expandable capacity, and reliability—have enabled the magnetostrictive delay line to displace magnetic drums and cores in the small digital computer.

The following Bibliography, which has been compiled from many technical publications in this country and abroad, covers a wide area in theory, design, and application. It is of interest to note that the second reference, which discusses the use of a magnetostrictive delay line in an aircraft navigation system, has served as the stimulus for almost all of the consequent research and development. The third reference, which is the first published account of the use of this type of delay line in a digital computer, is of historical importance in the computer art.

Bibliography

[1] G. Bradfield, "A new electro-acoustic transducer," *Electronic Engrg.*, vol. 20, pp. 74–78; March, 1948.
[2] E. M. Bradburd, "Magnetostrictive delay line," *Elec. Commun.*, vol. 28, pp. 46–53; March, 1951.
[3] R. Millership, R. C. Robbins, and A. E. DeBarr, "Magnetostriction storage systems for a high speed digital computer," *Brit. J. Appl. Phys.* (Correspondence), vol. 2, p. 30; October, 1951.
[4] R. L. Beurle, "Video signal integration using acoustic delay lines and dynamic magnetic storage, pp. 231–235, in "Comunication Theory," W. Jackson, Ed., Academic Press, London, Eng; 1953.
[5] H. Epstein and O. Stram, "Magnetostrictive sonic delay line," *Rev. Sci. Instr.*, vol. 24, pp. 231–232; March, 1953.
[6] R. C. Robbins and R. Millership, "Applications of magnetostriction delay lines," 1953 *Proc. Symp. on Automatic Digital Computation*, Natl. Phys. Lab., Teddington, Eng., pp. 199–212; 1954.
[7] A. E. DeBarr, "Digital storage using ferromagnetic materials," *The Elliot J.*, (Eng.), vol. 1, pp. 116–120; May, 1953.
[8] G. B. B. Chaplin, R. E. Hayes, and A. R. Owens, "A transistor digital fast multiplier with magnetostrictive storage," *Proc. IEE*, vol. 102, pt. B, pp. 412–425; July, 1955.
[9] A. Tarabella and R. Bibron, "A magnetostrictive delay line," *L'Onde Electrique*, vol. 35, pp. 682–686; July, 1955.
[10] A. Tarabella, "A decoder with a magnetostrictive delay line," *L'Onde Electrique*, vol. 35, pp. 985–989; November, 1955.
[11] G. G. Scarrott and R. Naylor, "Wire-type acoustic delay lines for digital storage," *Proc. IEE*, vol. 103, pt. B, suppl. no. 3, pp. 497–508; March, 1956.
[12] J. W. Fairclough, "A sonic delay-line storage unit for a digital computer," *Proc. IEE*, vol. 103, pt. B, suppl. no. 3, pp. 491–496; March, 1956.

* Received by the PGEC, December 12, 1960.

[13] J. A. Tempel, "The acoustic-delay-line electronic calculator," *Proc. IEE*, vol. 103, pt. B, suppl. no. 2, pp. 269–275; March, 1956.
[14] J. A. M. Lyon and T. B. Thompson, "Analysis and application of magnetostriction delay lines," IRE Trans. on Ultrasonics Engineering, no. PGUE-4, pp. 8–22; August, 1956.
[15] V. L. Johnson, "ATC transponder—the design and application of a magnetostriction delay line for coding purposes," IRE Trans. on Aeronautical and Navigational Engineering, vol. ANE-3, pp. 113–116; September, 1956.
[16] D. Maeder, "A decade pulse counting system with circulating storage in a delay line," *Helv. Phys. Acta*, vol. 29, pp. 459–462, FASC. 5–6; 1956.
[17] D. Maeder, "Build-up of large signals with elimination of reflections on magnetostrictive storage lines," *Z. angew. Math. u. Phys.*, vol. 8, pp. 326–327; July, 1957.
[18] D. Maeder, "Build-up of large signals with suppression of reflections in magnetostrictive storage lines by multicoil arrangements" *Helv. Phys. Acta*, vol. 30, pp. 347–352, FASC. 4; 1957.
[19] P. L. Dreyfus, H. G. Feissel, and B. M. Leclerc, "A magnetic drum extension to the Gamma 3 computer," 1956 IRE Convention Record, pt. 4, pp. 105–108.
[20] H. Epstein and O. B. Stram, "A high-performance magnetostriction-sonic delay line," IRE Trans. on Ultrasonics Engineering, no. PGUE-5, pp. 1–24; August, 1957.
[21] A. Rothbart and L. Rosenberg, "A theory of pulse transmission along a magnetostrictive delay line," IRE Trans. on Ultrasonics Engineering, no. PGUE-6, pp. 32–58; December, 1957.
[22] G. I. Cohn, L. C. Peach, M. Epstein, H. O. Sorenson, and D. P. Kanellakos, "Magnetostrictive delay line for video signals," IRE Trans. on Component Parts, vol. CP-5, pp. 53–59; March, 1958.
[23] D. Maeder, "A transistorized memory line," *Nuclear Instr.*, vol. 2, pp. 203–206; February, 1958.
[24] L. Rosenberg and A. Rothbart, "Electrical design of the transducer networks of a magnetostrictive delay line," 1958 IRE National Convention Record, pt. 2, pp. 92–101.
[25] V. N. Kostic, "Magnetostrictive delay line," *Bull. Instr. Nuclear Sci., Boris Kidrich* (Yugoslavia), vol. 7, pp. 97–101; March, 1957.
[26] R. C. Williams, "Theory of magnetostrictive delay lines for pulse and continuous wave transmission," IRE Trans. on Ultrasonics Engineering, no. PGUE-7, pp. 16–38; February, 1959.
[27] G. V. Glebovich, "On the pass-band of a magnetostrictive delay line," *Radiotekhnika* (*USSR*), vol. 14, pp. 74–75; February, 1959.
[28] H. H. Schiller, "Systems organization of a special purpose airborne digital computer," *Proc., 3rd Natl. Conf. on Military Electronics*, Washington, D. C., June 29–July 1, 1949, pp. 457–458.
[29] A. Rothbart, "A torsional magnetostrictive delay line," Proc. IRE (Correspondence), vol. 47, pp. 1153–1154; June, 1959.
[30] L. Packer, M. Raphael, and H. Saks, "Thirty-two aircraft radar track simulator," IRE Trans. on Military Electronics, vol. MIL-3, pp. 114–122; July, 1959.
[31] H. E. Vaughan, "Research model for time-separation integrated communication," *Bell Sys. Tech. J.*, vol. 38, pp. 909–932; July, 1959.
[32] D. A. Aaronson and D. B. James, "Magnetostrictive ultrasonic delay lines for a PCM communication system," IRE Trans. on Electronic Computers," vol. EC-9, pp. 329–332; September, 1960.
[33] S. J. Meyers, L. Rosenberg, and A. Rothbart, "A time-compressor using magnetostrictive delay lines," 1960 IRE International Convention Record, pt. 9, pp. 195–203.
[34] H. A. Showell, C. W. M. Barrow, and R. E. Collis, "Magnetostrictive delay-line stores operating at 1 Mc/s using transistor circuits," *Proc. IEE*, vol. 106, pt. B, suppl. no. 18, pp. 1267–1276; April, 1960.
[35] A. Rothbart, "A non-return to zero (NRZ) mode of operation for a magnetostrictive delay line," Proc. IRE (Correspondence), vol. 48, pp. 1486–1487; August, 1960.
[36] R. M. Beck, "PB 250—a high-speed serial general purpose computer using magnetostrictive delay line storage," *Proc. EJCC*. New York, N. Y., December, 1960, pp. 283–297.
[37] G. Svala, "Transducer properties in magnetostrictive delay lines," *Elteknik* (*Sweden*), vol. 2, pp. 92–95; June, 1959.

Author Index

Arthur Rothbart
98 Van Cortlandt Park South
Bronx 63, N. Y.

Analysis of a Crossed-Film Cryoton Shift Register*

Introduction

This article presents a new mode of a crossed-film cryotron (CFC) shift register which has been described in a previous paper.[1] In addition, the criteria of operation are analyzed, the time constant and the maximum speed of operation are calculated, and the experimentally observed outputs are presented.

Description

The shift register can be considered as a cascade of the CFC memory cell, shown in

* Received by the PGEC, January 31, 1961. An expanded version of this letter was presented at the Office of Naval Research Conf. on Superconductive Devices, Washington, D. C.; May, 1960.

[1] V. L. Newhouse, J. W. Bremer, and H. H. Edwards, "An improved film cryotron and its application to digital computers," Proc. IRE, vol. 48, pp. 1395–1404; August, 1960.

Fig. 1, which is formed by the parallel connection of the gate of one CFC and the control grid of a second CFC. Essentially all of I_0, the current injected at the junction of two parallel paths, will flow through gate *ab* since the inductance of the control grid is about 50 times that of the gate.

Pulsing i_2, the input grid current, will make gate *ab* resistive, and the gate current will be deflected to the superconducting control grid *d*. Turning off i_2 returns gate *ab* to the superconducting state, but the current is left flowing in control grid *d* since no resistance is present to force the current into the gate.

A circulating current can now be set up in the cell by switching off I_0. This is equivalent to injecting a current of $-I_0$ into the circuit. As before, all this injected current flows through gate *ab*, and the current in the grid *d* is unchanged. This leaves a circulating current flowing in a clockwise direction around the loop.

It is assumed that $I_0 < I_c$, the critical current of the gate. If this were not true, thermal effects might complicate the analysis; in any case, it would be possible to store current without pulsing i_2. This case is not of interest and will not be discussed.

The structure of the shift register is shown in Fig. 2. Information is transferred from left to right in the register in the form

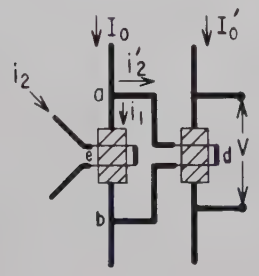


Fig. 1—The CFC storage cell.

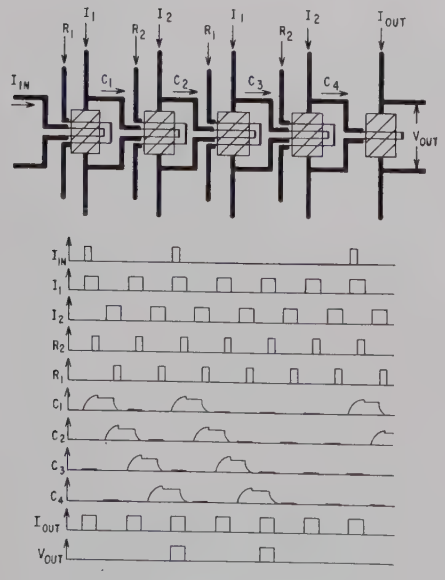


Fig. 2—Calculated waveforms for a four-loops-per-bit shift register having an input of 1, 1, 0, 1.

of stored currents. Current is stored in the first loop of the register in the same manner as described above for the memory cell. This stored current C_1 is used to switch the advance current I_2 injected into the second loop. After the stored current in the first stage is destroyed, the advance current of the second stage can be switched off, storing a current in that loop. This is done, as shown in Fig. 2, by an independent control grid, called a reset grid, crossing each gate. This is pulsed whenever it is desired to destroy the stored current in that loop of the register.

Binary information can be propagated in the register by defining a circulating current as a "1" (obtained by pulsing I_{in} and I_1 simultaneously), and the absence of a circulating current as a "0" (not pulsing I_{in} when I_1 is pulsed). If, as shown in Fig. 2, an output voltage pulse occurred simultaneously with the I_{out} pulse, a "1" was inserted into the shift register; if no voltage pulse occurred, a "0" was inserted into the register.

This shift register has at least two practical modes. One requires three loops per bit and three advance and three reset current sources. This mode has been described in a previous paper.[1] The second, although using four loops per bit, requires only two advance and two reset current sources, and is shown in Fig. 2. The advance currents are represented by *I*'s, the reset currents by *R*'s, and the control grid currents by *C*'s. The four-loop-per-bit mode is possible because information is inserted into the register on every second cycle of the advance drivers. If information were inserted on every cycle, as in the three-loop-per-bit mode, successive circulating currents would tend to destroy each other.

Analysis

A stored current which is propagated through a properly-operating circular shift register will reach one of two equilibrium values. One of these equilibrium values is zero. To calculate the other value in terms of the advance current, the reset current, and the CFC parameters, the following assumptions are made:

1) The advance current amplitude, I_0, is less than I_c.
2) All the CFC's of the register have identical characteristics. The reset and switching grids are assumed to have equivalent switching properties.
3) The amplitudes of all the advance currents are equal.
4) The amplitudes of all the reset currents are equal.
5) The ratio of grid inductance to gate inductance is infinite.

The equilibrium value of the circulating current can be determined from the CFC characteristic curve, $i_1 = f(i_2)$, which is shown schematically in Fig. 3. i_1 is the critical gate current that can flow with a grid current i_2 present. Let this CFC be part of the memory cell of Fig. 1, which is in a circular shift register. i_2 now becomes the circulating current of a previous loop of the register. After I_0 has been switched by i_2, i_1 is the unswitched current remaining in the gate and is given by

$$i_1 = f(i_2). \tag{1}$$

The switched current i_2' is, of course,

$$i_2' = I_0 - i_1. \tag{2}$$

Since an infinite inductance ratio has been assumed, i_2' will also be the stored current of this cell after i_2 and I_0 are switched off. Since i_2 and i_2' are both stored currents, the equilibrium value of a stored current in the register has been reached if

$$i_2 = i_2'. \tag{3}$$

If we substitute i_2 for i_2' in (2), the equilibrium value is determined by simultaneous solution of (2) and (3) for i_2.

A graphical solution is performed in Fig. 3. If (2) and (3) do intersect, two solutions, *A* and *B*, are obtained. If they do not intersect, there will be no nonzero equilibrium value of circulating current, and hence a "1" will not propagate until I_0 is increased sufficiently to allow intersection. If the curves do not intersect for any value of $I_0 < I_c$, then the CFC has insufficient current gain to act as a shift register.

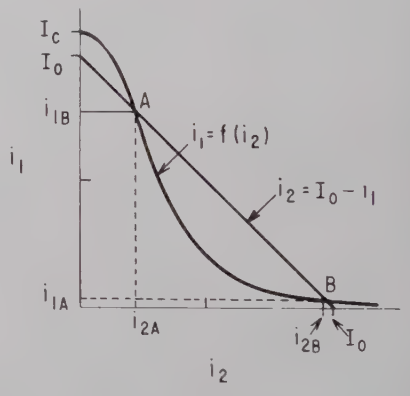


Fig. 3—Graphical solution for the equilibrium stored current in a shift register.

There are three regions of Fig. 3 where i_2, the input circulating current, can lie. Inspection of these regions reveals whether *A* and *B* are stable or unstable solutions. First, notice that if i_2 is the input circulating current of a cell, the stored current of that cell is $i_2' = I_0 - i_1$, where i_1 is determined by $i_1 = f(i_2)$. Considering the relation between i_2' and i_2 for the three regions:

1) If $i_2 < i_{2A}$, $i_2' < i_2$ and successive stored currents will tend to zero.
2) If $i_{2A} < i_2 < i_{2B}$, $i_2' > i_2$ and successive stored currents increase towards i_{2B}.
3) If $i_2 > i_{2B}$, $i_2' < i_2$ and successive stored currents decrease towards i_{2B}.

Thus, 0 and i_{2B} are the two stable solutions for the circulating current, i_{2B} resulting if

$$I_{in} > i_{2A}, \tag{4}$$

and 0 if

$$I_{in} < i_{2A}.$$

Also, the lower limit on the advance current, I_0, is the minimum value that allows intersection of curves 2) and 3).

The reset current must be large enough to make the gate resistive even at low values of gate current. Although lower values can be used, a practical upper limit is that current for which the portion of the gate directly under the grid is completely resistive. This is the saturation value of the gate resistance, and although the reset current could be safely increased much higher, this would serve no purpose since the gate resistance is essentially at its maximum value,[1] and hence the decay time of the stored current is minimized.

High-Speed Operation

The above analysis assumes that the current distribution has reached equilibrium when any of the current sources is switched on or off. It is usually adequate to allow 3τ for a stored current to switch an advance current, where τ is the L/R time constant of the circuit. An equal amount of time is sufficient for a reset current to destroy a stored current. Thus, the advance current only has to be switched on for a time 6τ—3τ to allow the stored current of the previous stage to switch the advance currents and 3τ to allow the reset current of the previous stage to destroy its stored current.

The value of τ calculated from the measured resistance and the calculated inductance of the circuit is 0.33 microsecond. This compares favorably with the accurately observed value[1] of 0.38 microsecond. An order of magnitude value for τ can also be obtained by observing the output signal of the storage cell of Fig. 1, which is shown in Fig. 4. The output signal was amplified by a Tektronix type-121 wideband preamplifier in cascade with a type-541 Tektronix oscilloscope. Amplifier oscillation is responsible for the noise.

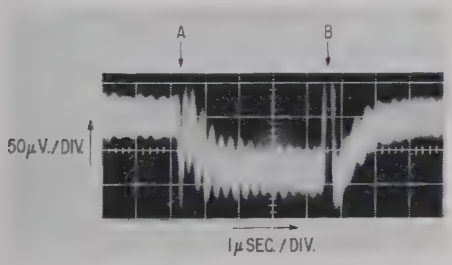


Fig. 4—Output voltage of a CFC. I_0 and I_0' (see Fig. 1) are on at the start of the trace. i_2 is switched on at A, switching I_0, and I_0 is switched off at B, allowing i_2 to destroy the stored current.

If we know τ, the highest speed of operation of the shift register can be calculated. Since the advance pulses can overlap each other in the three loops per bit mode, the minimum cycle time is 9τ, or 3.42 microseconds. This corresponds to an information rate of just under 300 kc.

All the testing was performed using the four-loops-per-bit mode of Fig. 2. Here the advance pulses cannot be overlapped and the minimum cycle time for the reset and advance pulse train is 12τ, or 4.56 microseconds, a rate of over 200 kc. The circuit has functioned perfectly at an advance pulse rate of 140 kc. In this experiment, the advance pulses were 3 microseconds long and the reset pulses were 1.2 microseconds long. All the current sources were 300 ma and the critical gate current was 350 ma. Higher speeds have not been attempted because of instability in the timing of the pulse equipment. The information rate in this mode is, of course, one half the advance pulse rate.

Although the register has only seven gates, longer registers can be formed by connecting the seven gate registers in cascade. This has been done successfully, but because of the unshielded interconnecting wires, the speed of operation is considerably reduced.

Output Waveforms

Fig. 5 shows the output pulses obtained by use of a 500-kc bandwidth, 50-db gain transistor pre-amplifier as an input to a 541 Tektronix oscilloscope.

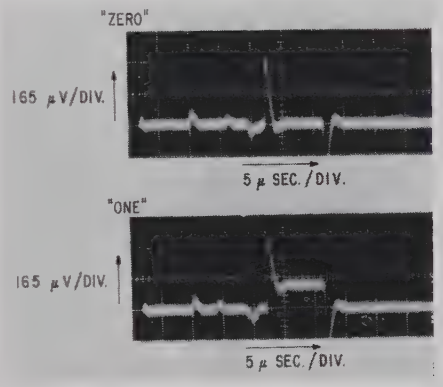


Fig. 5—Output waveforms of a CFC shift register.

The output gate of the register was pulsed with a 10-microsecond, 150-ma output pulse. The "ZERO" shows the output voltage with no circulating current in the last loop of the register. The noise was caused by radiation from the advance and reset pulses and the pulsing of the finite inductance of the output gate.

The "ONE" shows the 130-microvolt output signal caused by a stored current of 150 ma in the last loop. All the advance and reset pulses were 150 ma and the critical gate current was 200 ma.

Acknowledgment

The CFC shift register described in this letter and many, many others were produced by H. E. Tanis and D. MacKellar. Their contribution is gratefully acknowledged.

H. H. Edwards
General Electric
Light Military Electronics Dept.
Ithaca, N. Y.
V. L. Newhouse
General Electric Research Lab.
Schenectady, N. Y.
J. W. Bremer
General Electric Computer Dept.
Palo Alto, Calif.

A Note on Optimum Pattern Recognition Systems*

Chow[1] has shown that, for a given rejection rate, the error rate in a recognition system is minimized if the following decision criterion is used.

Choose class k if

$$p_k F(v \mid a_k) \geq p_j F(v \mid a_j) \qquad \text{for all } j \neq k$$

and

$$p_k F(v \mid a_k) \geq \beta \sum_{i=1}^{c} p_i F(v \mid a_i) \qquad 0 \leq \beta \leq 1;$$

reject the pattern if

$$p_j F(v \mid a_j) < \beta \sum_{i=1}^{c} p_i F(v \mid a_i) \quad \text{for all } 1 \leq j \leq c.$$

Here p_i is the *a priori* probability of the occurrence of class i, $F(v|a_i)$ is the conditional probability of making the measurement v given that a member of class i is present, c is the number of pattern classes and β is a constant chosen to force the system to meet the given rejection rate.

The proper value for β is generally difficult to determine, and an empirical approach may often be necessary. However, there is one important case in which β may be determined analytically, the discussion of which follows.

Let the cost of misrecognizing a pattern, of rejecting a pattern, and of correctly recognizing a pattern be independent of the pattern class. In particular, let

$$w_{ij} = w = \text{cost of misrecognition}$$
$$w_{io} = w_o = \text{cost of rejection}$$
$$w_{ii} = 0 = \text{cost of recognition}$$

where

$$w > w_o > 0.$$

(Since a Bayes criterion is being used, no generality is lost by setting $w_{ii}=0$.[2]) The general loss function is given by[1]

$$R(p, \delta) = \sum_{i=1}^{c} \sum_{j=0}^{c} \int_v \delta(d_j \mid v) p_i w_{ij} F(v \mid a_i) dv$$

where $\delta(d_j|v)$ is the probability that class j will be decided given the measurement v (d_o is the rejection decision), and $R(p, \delta)$ is the loss associated with the *a priori* class distribution p and a given decision function δ.

Using the above cost schedule, the loss function may be written

$$R(p, \delta) = \int_v \sum_{j=0}^{c} \delta(d_j \mid v) \sum_{i=1}^{c} w p_i (Fv \mid a_i) dv$$
$$- \int_v \sum_{i=1}^{c} \delta(d_i \mid v) w p_i F(v \mid a_i) dv$$
$$- \int_v \delta(d_o \mid v) \sum_{i=1}^{c} (w - w_o) p_i F(v \mid a_i) dv.$$

* Received by the PGEC, January 23, 1961.

[1] C. K. Chow, "An optimum character recognition system using decision functions," IRE Trans. on Electronic Computers, vol. EC-6, pp. 247–254; December, 1957.

[2] H. Chernoff and L. E. Moses, "Elementary Decision Theory," John Wiley and Sons, Inc., New York, N. Y., ch. 5; 1953.

Noting that

$$\sum_{j=0}^{c} \delta(d_j \mid v) = 1,$$

$$\sum_{i=1}^{c} p_i = 1,$$

$$\int_v F(v \mid a_i)dv = 1,$$

the first integral can be reduced, allowing the cost function to be written as

$$R(p, \delta) = w - w \int_v \sum_{i=1}^{c} \delta(d_i \mid v) p_i F(v \mid a_i)dv$$

$$- (w - w_o) \int_v \delta(d_o \mid v) \sum_{i=1}^{c} p_i F(v \mid a_i)dv.$$

To minimize $R(p, \delta)$, $\delta(d_i \mid v)$ is chosen as follows:

$$\delta(d_k \mid v) = 1, \qquad k \neq 0,$$

if

$$p_k F(v \mid a_k) \geq p_j F(v \mid a_j) \qquad \text{for all } j \neq k$$

and

$$p_k F(v \mid a_k) \geq \left(\frac{w - w_o}{w}\right) \sum_{i=1}^{c} p_i F(v \mid a_i);$$

$$\delta(d_o \mid v) = 1$$

if

$$p_j F(v \mid a_j) < \left(\frac{w - w_o}{w}\right) \sum_{i=1}^{c} p_i F(v \mid a_i)$$

for all $1 \leq j \leq c$.

But this decision criterion is of the same form as that derived by Chow for the case of minimum error rate given a fixed rejection rate, with

$$\beta = \frac{w - w_o}{w}.$$

Therefore, minimizing the cost in the case of constant costs also minimizes the error rate for the rejection rate which corresponds to the above β.

W. H. Highleyman
Bell Telephone Labs., Inc.
Murray Hill, N. J.

Minimal Characterizing Experiments for Finite Memory Automata*

In a recent article by Gill,[1] it has been shown that minimal characterizing experiments for automata of memory (length) M may be derived from a certain *sequential system*. In this note, this idea is carried slightly further so as to make more explicit the relationship between finite memory automata and the corresponding minimal characterizing experiments.

Consider the *maximal* machine of memory M, namely the one that actually contains 2^M states in its minimal Huffman flow table. As an example originally given by Simon,[2] Table I is the flow table of the maximal automaton of memory 2. One may convert this table into the Moore state transition table shown as Table II, by virtue of the following replacements:

$$0, a \to 0 \qquad 2, e \to 1$$
$$0, b \to 4 \qquad 2, f \to 5$$
$$1, c \to 2 \qquad 3, g \to 3$$
$$1, d \to 6 \qquad 3, h \to 7.$$

Now the transition diagram of the Moore model may be seen to yield precisely the oriented graph or *sequential system* of eight states illustrated in the Gill article.

TABLE I

	0	1
0	0, a	2, e
1	0, b	2, f
2	1, c	3, g
3	1, d	3, h

TABLE II

	0	1	
0	0	1	a
1	2	3	e
2	4	5	c
3	6	7	g
4	0	1	b
5	2	3	f
6	4	5	d
7	6	7	h

The above example illustrates, then, the following statement. The *cyclic minimal experiments* for the study of automata of memory M are all implicit in the transition diagram of the Moore model for the *maximal* automaton of memory M. As a matter of fact, these experiments may be read off the transition diagram as the sequences of input symbols associated with the arrows of directed, minimal, closed paths that traverse all the nodes or states.

Gill's results, as well as the contents of this note, seem to be direct consequences of Theorem 1 proved by Simon about finite memory automata.

J. E. Mezei
IBM Research
Yorktown Heights, N. Y.

* Received by the PGEC, February 23, 1961.

[1] A. Gill, "Characterizing experiments for finite-memory binary automata," IRE Trans. on Electronic Computers, vol. EC-9, pp. 469–471; December, 1960.

[2] J. M. Simon, "A note on the memory aspect of sequence transducers," IRE Trans. on Circuit Theory, vol. CT-6, pp. 26–29; March, 1959.

On the Size of Weights Required for Linear-Input Switching Functions*

A switching function $f(x_1, x_2, \cdots x_n)$ which takes on its values 0 and 1 according to a linear inequality,

$$f = 1 \Leftrightarrow \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n \geq \alpha_0,$$

is said to be a *linear-input* function,[1] also called linearly separated function,[2] threshold function,[3] setting function,[3] and in a more restricted sense, majority decision function.[4] The *weights* $\alpha_i (i = 1, 2, \cdots, n)$ and *threshold* α_0, which we may take to be integers without loss of generality, completely define any particular function of this type.

Several digital devices have been conceived which can realize any linear-input function with a single device. Although only a small fraction of all switching functions are linear-input functions, it is known that an arbitrary combinational switching function can be realized in a network of such devices. The potential simplicity of such networks in comparison with those composed of conventional digital elements justifies a more detailed study of this class of functions.

An important preliminary question concerns the maximum size of the weights α_i which may be required for the realization of arbitrary linear-input functions. In this note it is shown that one must be prepared for rather large weights, approaching a value which grows at a rate no less than $2^n/n$ for an n-input device, as n becomes large. This fact suggests in turn that if the nature of the device imposes any restriction on the maximum size of input weights, this will probably increase substantially the number of such devices required for the realization of arbitrary switching functions.

We lead up to this bound (Theorem 2) through a weaker theorem, but one whose proof is much simpler.

Theorem 1

Some linear-input functions of $n = 2q+1$ variables require weights at least as large as 2^q in any realization with integral weights.

Proof: Let a linear-input function f_1 be defined by

$$f_1 = 1 \Leftrightarrow \sum_{i=1}^{2q+1} \alpha_i x_i \geq \alpha_0,$$

in which the weights are

$$\alpha_1 = 1, \qquad \alpha_{q+2} = 2^q - 1,$$
$$\alpha_2 = 2, \qquad \alpha_{q+3} = 2^q - 2,$$
$$\alpha_3 = 4, \qquad \alpha_{q+4} = 2^q - 4,$$
$$\cdots\cdots \qquad \cdots\cdots\cdots$$
$$\alpha_q = 2^{q-1}, \qquad \alpha_{2q+1} = 2^q - 2^{q-1},$$
$$\alpha_{q+1} = 2^q, \qquad \alpha_0 = 2^q.$$

* Received by the PGEC, November 28, 1960; revised manuscript received, March 13, 1961.

[1] R. C. Minnick, "Linear-input logic," IRE Trans. on Electronic Computers, vol. 10, pp. 6–16; March, 1961.

[2] R. McNaughton, "Unate Truth Functions," Appl. Math. and Stat. Lab., Stanford University, Stanford, Calif., Tech. Rept. No. 4; October, 1957.

[3] M. C. Paull and E. J. McCluskey, Jr., "Boolean functions realizable with single threshold devices," Proc. IRE, vol. 48, pp. 1335–1337; July, 1960.

[4] S. Muroga, "Logical elements on majority decision principal and complexity of their circuit," *Proc. Internatl. Conf. on Information Processing*, Paris, France, June, 1959, UNESCO, Paris, 1960, Paper G.2.10, pp. 400–497; 1960.

Now consider any other realization of the same function f_1; let this realization be defined by

$$f_1 = 1 \Leftrightarrow \sum_{i=1}^{2q+1} \beta_i x_i \geq \beta_0.$$

For the same assignment of values to the variables $x_1, x_2, \cdots, x_n$, the pair of inequalities must be either both satisfied or both not satisfied. Thus, since

$$\alpha_{q+2} + \alpha_1 = \alpha_0 > \alpha_{q+2},$$

then

$$\beta_{q+2} + \beta_1 \geq \beta_0 > \beta_{q+2},$$

from which it follows that $\beta_1 > 0$, or $\beta_1 \geq 1$. Similarly, since

$$\alpha_{q+3} + \alpha_2 = \alpha_0 > \alpha_{q+3} + \alpha_1,$$

then

$$\beta_{q+3} + \beta_2 \geq \beta_0 > \beta_{q+3} + \beta_1,$$

so that $\beta_2 > \beta_1$, or $\beta_2 \geq 2$. Continuation of this process gives next

$$\beta_{q+4} + \beta_3 \geq \beta_0 > \beta_{q+4} + \beta_2 + \beta_1,$$

so that $\beta_3 > \beta_2 + \beta_1$, or $\beta_3 \geq 4$. In general, from

$$\beta_{q+j+1} + \beta_j \geq \beta_0 > \beta_{q+j+1} + \beta_{j-1} + \beta_{j-2} + \cdots + \beta_1, \quad (1)$$

we get

$$\beta_j > \beta_{j-1} + \beta_{j-2} + \cdots + \beta_1$$

or $\beta_j \geq 2^{j-1}$. This bound holds for $j = 2, 3, \cdots, q$. Finally,

$$\beta_{q+1} \geq \beta_0 > \beta_q + \beta_{q-1} + \cdots + \beta_1;$$

therefore, $\beta_{q+1} \geq 2^q$, establishing the desired bound.

The following theorem shows how this bound can be improved, through selection of a sequence of weights α_k which may be combined in various ways just barely to satisfy inequalities of the type used in Theorem 1.

Theorem 2

For sufficiently large n, and for any arbitrarily small positive constant δ, some linear-input functions of n variables require weights at least as large as

$$\left(\frac{2}{e} - \delta\right)\frac{2^n}{n}$$

in any realization with integral weights.

Proof: f_1 is now defined by the set of weights

$$\alpha_k = \begin{cases} 2^{k-1} \\ \alpha_{k-1} + \alpha_{k-2} + \cdots \alpha_{k-\nu} \\ \alpha_{k-\nu-1} \end{cases} \quad \begin{array}{l} 1 \leq k \leq \nu \\ \nu + 1 \leq k \leq n - \nu + 1 \\ n - \nu + 2 \leq k \leq n \end{array}$$

where ν is allowed to be any integer satisfying

$$1 < \nu \leq n/2,$$

and

$$\alpha_0 = \alpha_n - v + 1.$$

Thus, each weight α_k of the central group is set equal to the sum of the previous ν weights. The theorem will be proved by showing that: 1) one of the α_k, namely $\alpha_{n-\nu+1}$, is no smaller than the bound stated in the theorem; 2) any other realization of the same function has weights β_k which are no smaller than the corresponding α_k, up to $k = n - \nu + 1$.

The difference equation for the central group of α_k can be solved by well-known methods:[5] substitution of $\alpha_k = A\lambda^k$ yields

$$\alpha_k = \sum_{i=1}^{\nu} A_i \lambda_i^k,$$

where the A_i are constants depending on the boundary conditions (the first ν of the α_k), and the λ_i are the roots of

$$\lambda^\nu = \lambda^{\nu-1} + \lambda^{\nu-2} + \cdots + \lambda + 1.$$

Multiplication of both sides by a dummy factor $(\lambda - 1)$, and recombination of terms, gives

$$\lambda^\nu = \frac{1}{2 - \lambda},$$

which reveals that the polynomial has: 1) a real root λ_1 near $\lambda = 2$, approaching the value 2 monotonically from below as ν increases, and 2) ν other roots not far from the νth roots of unity, which are equally spaced around the unit circle in the λ plane.

It is not hard to show[6] that, except for the dummy root at $\lambda = 1$, all of these other roots lie *inside* the unit circle for $\nu < \infty$. Thus, all terms of the solution, except the first, diminish in size with increasing k. For sufficiently large k, therefore, we can make

$$\alpha_k \geq A_1 \lambda_1^k - \delta_1$$

for any given positive constant δ_1. Application of the boundary conditions (*e.g.*, through solution in terms of Vandermonde determinants)[7] gives

$$A_1 = \frac{1}{\lambda_1} \prod_{i=2}^{\nu} \left| \frac{2 - \lambda_i}{\lambda_1 - \lambda_i} \right| > \frac{1}{\lambda_1};$$

therefore for $k = n - \nu + 1$,

$$\alpha_{n-\nu+1} \geq \frac{1}{\lambda_1} \lambda_1^{n-\nu+1} - \delta_1 = \lambda_1^{n-\nu} - \delta_1 = (2 - \lambda_1)\lambda_1^n - \delta_1.$$

This form of the bound shows that as we try to increase ν to bring λ_1^n closer to 2^n, the first factor becomes diminishingly smaller. To achieve the greatest rate of growth, we may select ν to satisfy

$$2 - \lambda_1 = \frac{1}{\lambda_1^\nu} = \frac{2}{n},$$

so that

$$\alpha_{n-\nu+1} \geq \frac{2}{n}\left(2 - \frac{2}{n}\right)^n - \delta_1 = \frac{2^{n+1}}{n}\left(1 - \frac{1}{n}\right)^n - \delta_1.$$

The second factor approaches $1/e$ as $n \to \infty$; for sufficiently large n, we can make[8]

$$\left|\left(1 - \frac{1}{n}\right)^n - e^{-1}\right| \leq \delta_2/2$$

for any given positive constant δ_2, so that

$$\alpha_{n-\nu+1} \geq \frac{2^n}{n} \cdot 2\left(e^{-1} - \frac{\delta_2}{2}\right) - \delta_1$$

or

$$\alpha_{n-\nu+1} \geq \frac{2^n}{n}\left(\frac{2}{e} - \delta\right), \quad (2)$$

where

$$\delta = \delta_2 + \frac{n}{2^n}\delta_1.$$

Clearly, for given δ, we can select $\delta_2 < \delta$ and choose n sufficiently large to satisfy this equation.

We now modify the argument in the proof of Theorem 1 to show that

$$\beta_k \geq \alpha_k \qquad k = 1, 2, \cdots, n - \nu + 1.$$

The last inequality of this sequence, in conjunction with (2), will then establish the theorem.

The earlier proof required that we be able to form inequalities in the α_j corresponding to the inequalities like (1) in the β_j. Consider, therefore,

$$C_j + \alpha_j \geq \alpha_0 > C_j + \sum_{\substack{i=0 \\ i \not\equiv 0 \pmod{\nu}}}^{j-1} \alpha_{j-i}, \qquad j = 1, 2, \cdots, n - \nu + 1 \quad (3)$$

where[9]

$$C_j = \sum_{\substack{i=0 \\ i \not\equiv 0 \pmod{\nu}}}^{n-\nu} \alpha_{j+1} + \sum_{i=1}^{j-n-1 \pmod{\nu}} \alpha_{n-\nu+1+i}.$$

We note first that no α_j is repeated in these sums, so that the expressions correspond to selected values of the binary variables $x_1, x_2, \cdots, x_n$. We must now show that the inequalities (3) are satisfied by the set of α_j defined at the beginning of this proof.

For the central group of α_k, the difference expression

$$\alpha_k = \sum_{i=1}^{\nu} \alpha_{k-i}$$

[5] F. B. Hildebrand, "Methods of Applied Mathematics," Prentice-Hall, Inc., New York, N. Y., pp. 227–230; 1952.

[6] For example, the number of zeros of an analytic function, here $F(\lambda) = \lambda^\nu - \lambda^{\nu-1} - \cdots - \lambda - 1$, which lie inside the unit circle in the λ-plane equals the number of encirclings of the origin of $F(e^{i\theta})$ as θ ranges from 0 to 2π.

[7] *E.g.*, F. E. Hohn, "Elementary Matrix Algebra," The Macmillan Co., New York, N. Y., p. 47; 1958.

[8] K. Knopp, "Theory and Application of Infinite Series," Hafner Publishing Co., Inc., New York, N. Y., pp. 193–194; 1947.

[9] We use the convention that summation over indices outside of the indicated range yields 0; thus, $C_{n-\nu+1} = 0$.

may be substituted into its own last term $\alpha_{k-\nu}$ to get

$$\alpha_k = \sum_{i=1}^{\nu-1} \alpha_{k-i} + \sum_{i=\nu+1}^{2\nu} \alpha_{k-i}$$

and again into the last term $\alpha_{k-2\nu}$ of this sum, etc., to eliminate every νth term:

$$\alpha_k = \sum_{\substack{i=0 \\ i \not\equiv 0 \pmod{\nu}}}^{k-1} \alpha_{k-i} + 1.$$

The residual 1 arises from the last term of the final sum, which cannot be eliminated. We note that this form of the recursion also holds for the lower group of α_k, since

$$2^{k-1} = \sum_{i=1}^{k-1} 2^{i-1} + 1.$$

With $k=j$, then,

$$C_j + \alpha_j > C_j + \sum_{\substack{i=0 \\ i \not\equiv 0 \pmod{\nu}}}^{j-1} \alpha_{j-i}$$
$$j = 1, 2, \cdots, n-\nu+1.$$

and (3) is partially established.

We next show that $C_j+\alpha_j=\alpha_0$, completing the proof. Reversing the difference equation to

$$-\alpha_k = \sum_{i=1}^{k-1} \alpha_{k+i} - \alpha_{k+\nu} \quad 1 \le k \le n-2\nu+1,$$

we may repeatedly substitute this expression in its own last term, as before, to eliminate every νth term, but without going beyond the term $\alpha_{n-\nu+1}$:

$$-\alpha_k = \sum_{\substack{i=0 \\ i \not\equiv 0 \pmod{\nu}}}^{r} \alpha_{k+1} - \alpha_{k+r\nu},$$

where the integer r is selected according to

$$n-2\nu+1 < k+r\nu \le n-\nu+1. \quad (4)$$

From the original difference expression, we have

$$\alpha_0 = \alpha_{n-\nu+1} = \sum_{i=1}^{\nu} \alpha_{n-\nu+1-i},$$

so that, adding these two sums,

$$\alpha_0 - \alpha_k = \sum_{\substack{i=0 \\ i \not\equiv 0 \pmod{\nu}}}^{r\nu} \alpha_{k+i} - \alpha_{k+r\nu} + \sum_{i=1}^{\nu} \alpha_{n-\nu+1-i}.$$

While the indices in the left-hand sum increase from k, those in the right-hand sum decrease from $n-\nu$. The two sums overlap, and the right-hand sum contains within it the term $\alpha_{k+r\nu}$:

$$\alpha_0 - \alpha_k = \sum_{\substack{i=0 \\ i \not\equiv 0 \pmod{\nu}}}^{r\nu} \alpha_{k+1} - \alpha_{k+r\nu}$$
$$+ (\alpha_{n-\nu} + \alpha_{n-\nu-1} + \cdots + \alpha_{k+r\nu+1})$$
$$+ \alpha_{k+r\nu} + (\alpha_{k+r\nu-1} + \cdots + \alpha_{n-2\nu+1}).$$

The terms in the first parentheses constitute a direct continuation of the first sum. The terms in the second parentheses are $(k+r\nu)-(n-2\nu+1) \equiv k-n-1 \pmod{\nu}$ in number, and may be collected in a second sum:

$$\alpha_0 - \alpha_k = \sum_{\substack{i=0 \\ i \not\equiv 0 \pmod{\nu}}}^{n-\nu} \alpha_{k+i} + \sum_{i=1}^{k-n-1 \pmod{\nu}} \alpha_{n-2\nu+i}.$$

But $\alpha_k=\alpha_{k-\nu-1}$ for $k \ge n-\nu+2$; therefore,

$$\alpha_0 - \alpha_k = \sum_{\substack{i=0 \\ i \not\equiv 0 \pmod{\nu}}}^{n-\nu} \alpha_{k+i} + \sum_{i=1}^{k-n-1 \pmod{\nu}} \alpha_{n-\nu+1+i}$$
$$= C_k$$

for $k=1, 2, \cdots, n-2\nu-1$. When $r=0$, the first sum vanishes by (4), but the decomposition of the other terms is still valid Thus, this expression holds up to $k=n-\nu+1$:

$$C_k + \alpha_k = \alpha_0, \quad k = 1, 2, \cdots, n-\nu+1,$$

which completes the proof of Theorem 2.

The special case $\nu=2$ leads to the weight sequence 1, 2, 3, 5, 8, 13, 21, $\cdots$, which is the "Fibonacci series."[10] The general term arises from $\lambda^2=\lambda+1$, and is

$$\alpha_{n-1} = \frac{1}{\sqrt{5}} \left\{ \left(\frac{1+\sqrt{5}}{2} \right)^{n+1} - \left(\frac{1-\sqrt{5}}{2} \right)^{n+1} \right\},$$

leading to the bound $\beta_{n-1} \ge \alpha_{n-1}$.

It is not known whether this bound of $2^n/n$ can be improved asymptotically. For small n, it can certainly be improved. For $n=7$, for example, the proofs of Theorems 1 and 2 give bounds of 8 and (for $\nu=2$ and $\nu=3$) 13, respectively, whereas the following general number-theoretic statement of the problem yields for $n=7$ a linear-input function with a coefficient which cannot be reduced below 18:

Determine a sequence of integers α_1, α_2, $\cdots$, α_n and α_0 satisfying the following conditions:

1) $\alpha_1=1$, $\alpha_k > \alpha_{k-1}$, $\alpha_k > \sum L_k \alpha_j$, where L_k is a subset of α_i for which $i<k$,
2) $\alpha_0 - \alpha_k \le \sum R_k \alpha_i$, where R_k is a subset of α_i not including α_k, and not including any member of L_k, for $k=2, 3, \cdots p<n$. We seek to maximize α_n, which is identified with the bound.

The justification of this statement of the problem is contained implicitly in the proofs of the previous theorems.

The method of proof of the above theorems was devised by Myhill, while Kautz is responsible for its application and this presentation.

ACKNOWLEDGMENT

The authors would like to thank C. M. Ablow of Stanford Research Institute for assistance in the proof of Theorem 2.

J. MYHILL
Stanford University,
Stanford, Calif.
and Consultant
Stanford Research Inst.
Menlo Park, Calif.

W. H. KAUTZ
Stanford Research Inst.
Menlo Park, Calif.

[10] *Ibid.*, p. 14.

A Note on Moore's Distinguishability Theorem*

This note is concerned with the extension of Moore's Theorem 7, which appears in his basic paper "Gedanken Experiments on Sequential Machines."[1] The theorem, in its extended version, states:

> If M_1 is an n_1-state machine and M_2 an n_2-state machine, such that some state q_i of M_1 is distinguishable from state q_j of M_2, then the bound on the length L of the distinguishing experiment is given by:
>
> $$L \le n_1 + n_2 - 1 \quad (1)$$
>
> where the upper bound is achievable with equality.

Moore's Theorem 7 is the above theorem with $n_1=n_2=n$. The proof to (1) is the same as that to Theorem 7, with the trivial difference that the sum-machine to be considered has n_1+n_2 instead of $2n$ states. However, the proof that (1) is achievable with equality is not trivial, and requires some preliminary results that are of interest in their own right.

We shall start by introducing the following definitions:

(q_l, O) is said to be the *transition* of (q_k, I) if the state q_k, when presented with the input symbol I, yields the state q_l and the output symbol O. An experiment is said to *execute* the transition (q_l, O) of (q_k, I) if it takes the machine into state q_k and presents it with the input symbol I.

With reference to machines M_1 and M_2 of the theorem, a transition (q_l, O) in M_1 or M_2 is called a *critical transition*, if a change in q_l and/or O causes q_i to become equivalent to q_j.

A *singular machine* is one in which 1) the output alphabet contains only two symbols, O_1 and O_2; 2) exactly one transition, called the *singular transition*, contains the output symbol O_1 (or O_2).

In terms of the above definitions, the following lemma will now be presented:

> Let M_1 and M_2 be singular machines, where M_1 contains a critical transition. Let E be the shortest experiment which distinguishes state q_i of M_1 from state q_j of M_2. Then: 1) E executes the critical transition of M_1; 2) E terminates by executing the singular transition of M_1 or M_2.

Proof: Assuming that E does not execute the critical transition of M_1, the response of M_1 to E is insensitive to changes in this transition, and is, therefore, identical to that observed when q_i and q_j are equivalent. Since E is a distinguishing sequence, the assumption must be false, which proves part 1. E must execute the singular transition of M_1 or M_2 at least once, since otherwise the responses of M_1 and M_2 to E would be identical throughout. Also, any response to E which does not result from the execution of

* Manuscript received by the PGEC, February 9, 1961; revised manuscript received, March 18, 1961. Supported by the Information Systems Branch of the ONR under contract Nonr 222(53).

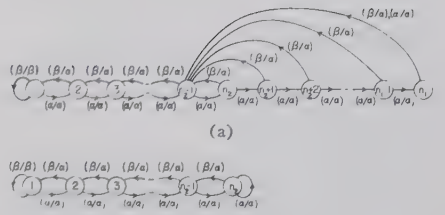
[1] E. F. Moore, "Gedanken-experiments on sequential machines," in "Automata Studies," Princeton University Press, Princeton, N. J., pp. 129–153; 1956.

TABLE I
MACHINE M_1

s_t \ x_t	s_{t+1} α	s_{t+1} β	y_t α	y_t β
1	2	1	α	β
2	3	1	α	α
3	4	2	α	α
4	5	3	α	α
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
n_2-1	n_2	n_2-2	α	α
n_2	n_2+1	n_2-1	α	α
n_2+1	n_2+2	n_2-1	α	α
n_2+2	n_2+3	n_2-1	α	α
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
n_1-2	n_1-1	n_2-1	α	α
n_1-1	n_1	n_2-1	α	α
n_1	n_2-1	n_2-1	α	α

TABLE II
MACHINE M_2

s_t \ x_t	s_{t+1} α	s_{t+1} β	y_t α	y_t β
1	2	1	α	β
2	3	1	α	α
3	4	2	α	α
4	5	3	α	α
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
n_2-1	n_2	n_2-2	α	α
n_2	n_2	n_2-1	α	α



(a)

(b)

Fig. 1—(a) Machine M_1. (b) Machine M^2.

one of the singular transitions is identical in M_1 and M_2, and hence cannot convey any additional information to the experimenter. Identification of q_i or q_j, therefore, is possible only when the singular transition of M_1 or M_2 is executed, which proves part 2 of the lemma.

Consider now the machines M_1 and M_2 specified by the state Tables I and II, respectively, and by Fig. 1. The specification follows Mealy's model,[2] with

$$s_{t+1} = f(s_t, x_t),$$
$$y_t = g(s_t, x_t),$$

where x_t, y_t and s_t are, respectively, the input symbol, the output symbol, and the state at time t. As previously shown,[3,4] all results obtained for Mealy's model are directly applicable to Moore's model (and vice versa). The labels (I/O) in Fig. 1 refer to the input-output pairs associated with the interstate transitions. $\{\alpha, \beta\}$ is the input and output alphabet of both machines, and it is assumed that $n_1 > n_2$.

M_1 and M_2 are seen to be singular, with the transition $(1, \beta)$ of $(1, \beta)$ being the singular transition in both machines. Also, the transition (n_2-1, α) of (n_1, α) in M_1 is critical when $q_i = q_j = 1$, which can be shown as follows: Replace the entry "n_2-1" in column α of row n_1 with the entry "n_1"; as a result, states n_1 and n_1-1 of M_1 become equivalent, and can be represented by the same row n_1-1 with the α entry n_1-1 and β entry n_2-1. At this point states n_1-1 and n_1-2 are seen to be equivalent, and the same merging process previously carried out on rows n_1 and n_1-1, can now be carried out on rows n_1-1 and n_1-2. The process can be continued until states n_2, n_2+1, $\cdots$, n_1 are all represented by the single row n_2, with the α entry n_2 and the βn_2-1. In this form, Table I becomes identical to Table II, which implies that state $k (1 \leqq k \leqq n_2)$ of M_1 is equivalent to state k of M_2. The transition (n_2-1, α) of (n_1, α) in M_1 is, therefore, proved to be critical when $q_i = q_j = 1$.

By the lemma, we can conclude that the minimal experiment employed for distinguishing state 1 of M_1 from state 1 of M_2 must execute the transition (n_2-1, α) of (n_1, α) in M_1. This requires at least n_1 input symbols (all α's), which take M_1 into state n_1-1 and M_2 into state n_2. Also by the lemma, we can conclude that the experiment must terminate by executing the transition $(1, \beta)$ of $(1, \beta)$ in either M_1 or M_2. This requires at least n_2-1 additional input symbols (all β's) which execute the transition $(1, \beta)$ of $(1, \beta)$ in M_1, and the transition $(1, \alpha)$ of $(2, \beta)$ in M_2.

Thus, the minimal experiment which distinguishes states 1 of M_1 and M_2 is of length n_1+n_2-1, which proves that (1) is achievable with equality.

ARTHUR GILL
Elec. Engrg. Dept.
University of California
Berkeley, Calif

[2] G. H. Mealy, "A method for synthesizing sequential circuits," *Bell Sys. Tech. J.*, vol. 34, pp. 1045–1079; September, 1955.
[3] W. J. Gadden, "Equivalent sequential circuits," IRE TRANS. ON CIRCUIT THEORY, vol. CT-6. pp. 30–34; March, 1959.
[4] A. Gill, "Comparison of finite-state models," IRE TRANS. ON CIRCUIT THEORY, vol. CT-7, pp 178–179; June, 1960.

CORRECTION

Games that Teach the Fundamentals of Computer Operation*

DOUGLAS C. ENGELBART†

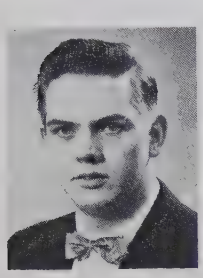
Reprints of this paper are available without charge directly from the author and not from the source listed at the head of the paper. The *Editor* regrets his error in listing the incorrect source.

—H. E. Tompkins, *Editor*

* IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-10, pp. 31–41; March, 1961.
† Stanford Research Institute, Menlo Park, Calif.

Contributors

Charles Avery Allen (S'56–M'57) was born in Bairdford, Pa., on April 25, 1933, He received the B.S.E.E. degree from the University of Pittsburgh, Pittsburgh, Pa., in 1956, and is presently studying for the M.S.E.E. degree under the IBM Fellowship Program at the California Institute of Technology, Pasadena.



C. A. Allen

In 1956 he joined the IBM Corporation at the Poughkeepsie, N. Y. Labratories, where he worked in Research on a new memory technology and later in the Data Systems Division on solid-state memory Development.

Mr. Allen is a member of Eta Kappa Nu and Sigma Tau.

❖

David R. Bennion (S'50–A'51–M'56) was born on February 13, 1929, in Salt Lake City, Utah. He received the B.S., M.S., and Ph.D. degrees in electrical engineering at Stanford University, Stanford, Calif., in 1951, 1953, and 1956, respectively.

D. R. Bennion

During the academic year 1951–1952 he attended the Oak Ridge School of Reactor Technology, Oak Ridge, Tenn., where he led a group assigned to a feasibility study of one type of nuclear reactor. He returned to Stanford in 1952 on a Tau Beta Pi fellowship. Specializing in communication theory, he continued graduate work for two additional years on a National Science Foundation fellowship. He then worked for a year as a research assistant at Stanford Electronics Laboratories, Stanford, Calif., during which time he completed his doctoral dissertation.

He joined the staff of Stanford Research Institute, Menlo Park, Calif., as a Research Engineer with the Devices and Circuits Group in July, 1956. After assisting in the final stages of the ERMA project, he became engaged in research on magnetic multi-aperture devices for computers. Since 1958, he has served as technical leader of research projects on various aspects of the general problem of all-magnetic logic.

Dr. Bennion is a member of the Scientific Research Society of America, Tau Beta Pi, and Phi Beta Kappa.

Sammy A. Butler (S'56–M'57) was born in Mount Sterling, Ill, on May 5, 1931. He received the B.S.E.E. degree from the University of Illinois, Urbana, in 1956.

S. A. Butler

In August, 1956, he joined the IBM Research Laboratory in Poughkeepsie, N. Y., where his initial assignment was in the area of magnetic device design for switching circuits. At present he is with the Advanced Systems Development Division of IBM, Yorktown Heights, N. Y., where he has been working on exploratory devices and circuits.

Mr. Butler is a member of Sigma Tau and Chi Gamma Iota.

❖

George D. Bruce (A'52–M'57) was born in Winter Haven, Fla., on January 8, 1926. He received the B.S. degree in physics and electrical engineering from the University of Pittsburgh, Pittsburgh, Pa., in 1950, and is presently completing the M.S.E.E. degree in the Syracuse University Graduate Program at IBM, Poughkeepsie, N. Y.

G. D. Bruce

In 1950 Mr. Bruce joined the IBM Corporation, and since 1952 he has been at the Poughkeepsie Laboratories, where he has worked on transistor switching circuits and on early solid-state computer development. Since 1958, he has been managing a memory development group and presently an advanced core memory system group in the Data Systems Division.

Mr. Bruce is a member of Sigma Pi Sigma and Sigma Tau.

❖

Warren A. Christopherson (S'50–A'52–M'57) was born in Benson, Minn., on May 20, 1922. He received the B.S.E.E. degree (with honor) from the California Institute of Technology, Pasadena, in 1948, and the M.S.E.E. and Ph.D. degrees in 1950 and 1952, respectively, from Stanford University, Palo Alto, Calif.

During 1948–49 he was employed at the General Electric Company, Schenectady, N. Y., for three engineering assignments. While a graduate student at Stanford he worked on the design of UHF amplifiers and served as a laboratory instructor. In 1952 he joined IBM in San Jose, Calif., where he is presently employed as a senior engineer. From 1952 to 1955 he was engaged in the development of data-gathering and data-reduction systems, magnetic-core storage, photographic storage, and optical mark sensing. During 1955 he served in a special assignment at IBM Headquarters in New York City, N. Y., as assistant to the director of advanced development. In 1956 he returned to IBM in San Jose and spent three years in the development of data-transmission systems. Since 1959 he has been developing magnetic-core storage for unit-record data processing systems.

W. A. Christopherson

Dr. Christopherson is a member of Tau Beta Pi and Sigma Xi.

❖

John T. Chu received the Ph.D. degree in mathematics from Iowa State University, Iowa City. He was a research associate in mathematical statistics at the University of North Carolina, Chapel Hill. From 1955 to 1957 he taught at Case Institute of Technology, Cleveland, Ohio. Before he joined the faculty of the Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, in 1960, he was a research mathematician at R.C.A. in Camden, N. J. His main interests are mathematical statistics, mathematical analysis, and switching logic.

J. T. Chu

❖

Edwin D. Councill (S'54–M'56) was born in Franklin, Va., on March 12, 1931. He received the B.S.E.E. degree from the University of Tennessee, Knoxville, in 1955, and is now participating in the Syracuse University Graduate Program at IBM, Poughkeepsie, N. Y.

In 1956, he joined the IBM Research Laboratory, Poughkeepsie, where he worked on magnetic storage devices and techniques.

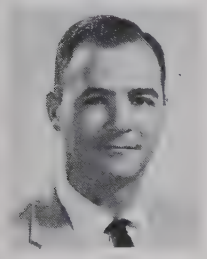
He entered the Product Development Laboratory, Data Systems Division in 1957, as a member of the STRETCH Memory Development Group. He is a Development Engineer in Memory Development, and is manager of Memory Machine Development.

E. D. Council

Mr. Council is a member of Tau Beta Pi, Eta Kappa Nu, and Phi Kappa Phi.

❖

Hewitt D. Crane (S'47–A'48–M'55–SM'57) was born in Jersey City, N. J., on April 27, 1927. He received the B.S. degree from Columbia University, New York, N. Y., in 1947, and the Ph.D. degree from Stanford University, Stanford, Calif., in 1960, both in electrical engineering.



H. D. Crane

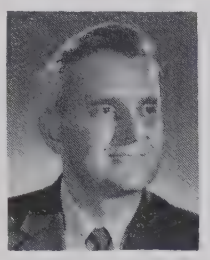
During 1945 and 1946, he served in the U. S. Navy. During 1947 to 1949, he designed inter-office communication equipment for the David Bogen Company in New York, N. Y., and did development work on facsimile equipment for the Western Union Telegraph Company, New York, N. Y. For the next two and one-half years, his work involved maintenance of IBM's SSEC computer in New York. From 1951 through 1954, at the Institute for Advanced Study at Princeton, N. J., he was concerned with the over-all logic of the IAS Computer and was in charge of designing and building the input-output equipment, a graphing device, and a magnetic secondary storage device. For one year at the David Sarnoff Research Center at Princeton, he worked on magnetic devices, in particular on multi-aperture magnetic devices and a large-scale magnetic memory.

He joined the staff of Stanford Research Institute in 1956. During his first six months at the Institute he was concerned with an analysis of the ERMA logic. Subsequently, he initiated work on the development of techniques for the realization of digital logic circuits with all-magnetic circuitry—in particular, using MAD's (multi-aperture magnetic devices). Over twenty inventions in the magnetic devices field have resulted in patent applications. More recently, he has initiated a program of study on a new class of computer devices termed Neuristors.

Dr. Crane is a member of Tau Beta Pi and Sigma Xi.

❖

Douglas C. Engelbart (S'51–M'56) was born on January 30, 1925, in Portland, Ore. He received a B.S.E.E. degree from Oregon State College, Corvallis, in 1948. In 1953, he received the M.S.E.E. degree and, in 1955, the Ph.D. degree in electrical engineering, from the University of California, Berkeley, While studying at the University of California, he was an Associate in Electrical Engineering. He became an Assistant Professor in 1955–1956.



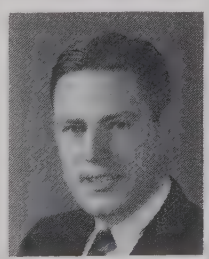
D. C. Engelbart

From 1948 to 1951, Dr. Engelbart was an Electrical Engineer in the Electrical Section of the Ames Laboratory at Moffet Field, Calif. In 1955–1956 he was a consultant to Marchant Research, Inc., Oakland, Calif. In 1956, he formed and directed a corporation, Digital Techniques, Inc., Berkeley, Calif., which, in 1956–1957, did further development work on his inventions. In 1957, he joined the staff of Stanford Research Institute, Menlo Park, Calif., where he has been concerned with basic developmental work on magnetic components for computers and with information retrieval problems. His fields of specialty include circuits special components, logical design, and programming of digital computers; vacuum and gas-discharge techniques; large inter-communication systems; wind tunnel drive and control systems; electro-mechanical control systems; and information retrieval systems.

Dr. Engelbart is a member of Pi Mu Epsilon, Sigma Tau, Tau Beta Pi, Phi Kappa Phi, Sigma Xi, Eta Kappa Nu.

❖

Herbert Freeman (A'49–M'52–SM'54) was born in Frankfurt-am-Main, Germany, on December 13, 1925. He received the B.S.E.E. degree from Union College, Schenectady, N. Y., in 1946, and the M.S., and Dr. Eng. Sc. degrees in electrical engineering from Columbia University, New York, N. Y., in 1948 and 1956, respectively.



H. Freeman

He joined the Sperry Gyroscope Company, Great Neck, N. Y., in June, 1948, as a project engineer. In February, 1957, he was promoted to head of the advanced studies department, and in December, 1959, to head of the military data-processing department. During the 1958–59 academic year, he served as visiting professor in the department of electrical engineering at Massachusetts Institute of Technology, Cambridge. In September, 1960, he joined the department of electrical engineering at New York University, New York, N. Y. He has been especially active in the field of control systems and special-purpose digital computers.

Dr. Freeman is a Registered Professional Engineer of New York State and a member of the AAAS, the ACM, the AIEE, Sigma Xi, and the New York Academy of Science.

❖

Juris Hartmanis was born in Riga, Latvia, on July 5, 1928. He received the Cand.Phil. degree in Physics from the University of Marburg, Germany, in 1949; the M.A. degree in Mathematics from the University of Kansas City, Mo., in 1951, and the Ph.D. degree in Mathematics from the California Institute of Technology, Pasadena, in 1955.

J. Hartmanis

From 1951 to 1955, he was a teaching assistant at Cal Tech; from 1955 to 1957, an instructor in the Mathematics Department of Cornell University, Ithaca, N. Y.; and from 1957 to 1958, an assistant professor at Ohio State University, Columbus. Since 1958, he has been a research mathematician in the Information Studies Section of the General Electric Research Laboratory, Schenectady, N. Y. He has done research in modern algebra and switching theory, and is now engaged in the study of the structure theory of sequential machines and the application of algebraic methods to this area of research.

Dr. Hartmanis is a member of the American Mathematical Society, the American Mathematical Association and Sigma Xi.

❖

John L. Haynes was born in Washington, D. C., on March 25, 1934. He received the B.E.E. degree from Cornell University, Ithaca, N. Y., in 1956 and the M.S.E.E. from Stanford University, Stanford, Calif., in 1958. While attending Cornell, he was employed part-time in the Cornell Aeronautical Laboratory in guided missile test and development. Subsequently he became an Instructor in the Electrical Engineering Department.

J. L. Haynes

In 1956, he joined the staff of Stanford Research Institute, where he was initially engaged in the analysis of antennas and radiation from airframes. He has since been concerned with high-density magnetic recording, reliability studies of mechanical assemblies under extreme environmental conditions, and reliability studies of a military reconnaissance system involving digital

transistor circuitry. More recent work involved a detailed analysis of the known physical means for realizing digital logic, and a basic investigation of digital computer memory access schemes.

Mr. Haynes is a member of Mu Sigma Tau, Eta Kappa Nu, and Tau Beta Pi.

❖

Herman J. Heijn was born in Zaadam, The Netherlands, on February 4, 1928. He studied at the Institute of Technology, at Delft, where he received his Engineering Degree in 1950, and his Doctor's Degree on the subject of "Representation of Switching Functions and their Application to Computers," in 1960.

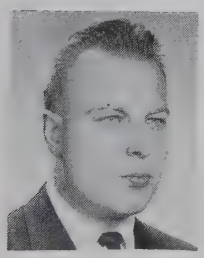
H. J. Heijn

After a one-year assistantship in the department of theoretical electricity of that institute, he joined the Philips Research Laboratories, Eindhoven, The Netherlands, in 1951. In the Computer Group of those laboratories, he took an active part in the design and the realization of the PASCAL and earlier experimental computers.

Dr. Heijn is a member of the "Nederlands Rekenmachine Genootschap."

❖

Reginald A. Kaenel was born in Berne, Switzerland, on October 22, 1929. He attended the Swiss Federal Institute of Technology (ETH) in Zurich, Switzerland, from which he received the Diploma in electrical engineering in 1955, and the Dr.Sc.Techn. degree in 1958 by a study of the use of magnetic square loop cores in pulse position modulation systems.



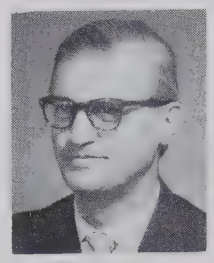
R. A. Kaenel

From 1956 to 1958, he worked as an assistant to and collaborator with the Professor of the Institute for Telecommunication at the ETH. In 1958, he joined the Bell Telephone Laboratories, Murray Hill, N. J., where he is currently concerned with Interference Analysis in the Systems Engineering Department.

Dr. Kaenel is a member of Commission IV of URSI.

❖

Ernst Kettel was born in Stadtkyll, Germany, on June 19, 1910. He received the Dipl.-Ing. degree in physics from Technische Hochschule, Aachen, Germany, in 1934, and the Dr. Ing. degree from the Technische Universität, Berlin, in 1958.



E. Kettel

Since 1934 he has been associated with Telefunken at Berlin. First he worked in the broadcast and television department and later he was engaged in radar problems. His main interest lies in systems and communication theory. At present he is a member of Telefunken Research Institute at Ulm, Germany. One of his groups is working in the field of analog computers.

Dr. Kettel is a member of the German Communication Engineering Society (NTG).

❖

Meir M. Lehman was born in Kralsruhe, Germany, in 1925 and emigrated to England in 1931. From 1941 to 1950, he was employed by Murphy Radio Ltd., Welwyn Garden City, Hertfordshire. In 1950 he was awarded a Ministry of Education Scholarship and read Mathematics at the Royal College of Science, Imperial College, London University, where he obtained the B.Sc. (hon.) degree in 1953. He then was awarded the Ferranti Research Scholarship of the Institution of Electrical Engineers and worked in the digital computer field. For this he obtained the Ph.D. degree from London University in 1957.

M. M. Lehman

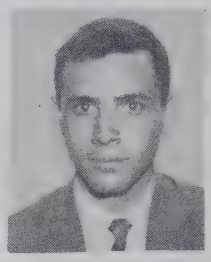
From 1956 to 1957, he was on the staff of the London Computer Laboratories, Ferranti Ltd. Since then he has been with the Scientific Department, Israel Ministry of Defence, Tel Aviv, as a "Foreign Expert."

In his capacity as a Senior Lecturer (part time) at the Technion, Israel Institute of Technology, he is conducting a graduate course on the systems and logical design of digital computer systems, and supervises graduate research students in the digital computer field.

Dr. Lehman is an Associate Member of the IEE, and a member of the British Computer Society, the ACM, and the Israel Mathematical Society.

❖

Kent V. Mina was born in Teaneck, N. J., on August 9, 1936. He received the B.S.E.E. degree from Princeton University, Princeton, N. J., in 1958, and the M.S.E.E. degree from New York University, New York, N. Y., in 1960.



K. V. Mina

He has been employed by Bell Telephone Laboratories, Murray Hill, N. J., since 1958, where he is completing the Communications Development Training Program and is currently associated with the Communications Techniques Research Department.

Mr. Mina is a member of Eta Kappa Nu.

❖

Edmund E. Newhall (S'49–A'51) was born in Calgary, Alberta, Canada on May 26, 1927. He received the B.S. degree in engineering physics from the University of Alberta in 1949, and the Ph.D. degree from the University of Toronto, Ontario, Canada, in 1958.

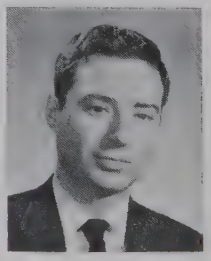
E. E. Newhall

From 1949 to 1952, he was associated with the Westinghouse Electric Company, Radar Division, Hamilton, Ontario. From 1952 to 1958, he taught electronic circuits at the University of Toronto, after which he joined the technical staff of Bell Telephone Laboratories, Murray Hill, N. J. He has been engaged in research of ferrite devices for memory and logic circuits and has written several technical articles.

Dr. Newhall is a member of the AIEE.

❖

Roy G. Saltman (S'52–M'55) was born in New York, N. Y., on July 15, 1932. He received the B.E.E. degree from Rensselaer Polytechnic Institute, Troy, N. Y., in 1953, and the M.S.E.E. degree from the Massachusetts Institute of Technology, Cambridge, Mass., in 1955.



R. G. Saltman

Since 1955, he has been employed by the Surface Armament Division, Sperry Gyroscope Company, Division of Sperry Rand Corporation, Great Neck, L. I., N. Y., and has held the title of senior engineer since July, 1960.

While at Sperry, his work has included the logical design, programming, and systems design of digital computers and the synthesis of military systems involving digital computers.

Mr. Saltman is a member of Eta Kappa Nu, Tau Beta Pi, and the ACM.

❖

Wolfgang Schneider was born in Ehingen am Danube, Germany, on August 11, 1929. He received the Dipl.-Ing. degree in electrical and light engineering from Technische Hochschule, Karlsruhe, Germany, in 1955.

W. Schneider

He joined the Telefunken Research Institute at Ulm, Germany, in 1956, where he has been working as a research engineer in the field of electronic analog computers.

❖

Johan C. Selman was born at Medan, Sumatra, on August 6, 1929. He studied at the Institute of Technology at Delft, The Netherlands, where he received his Engineering Degree in 1952.

J. C. Selman

In the same year he joined the Philips Research Laboratories, Eindhoven, The Netherlands. In the Computer Group of those laboratories he took part in the experiments during the design of the PASCAL computer, and he later directed its testing. Temporarily, he is also a lecturer on computer subjects at the Philips International Institute.

Mr. Selman is a member of the "Nederlands Rekenmachine Genootschap."

❖

William G. Strohm (S'55–M'57) was born in Rockville Centre, N. Y., on August 23, 1930. He received the B.E.E. degree in 1956 from the Rensselaer Polytechnic Institute, Troy, N. Y.

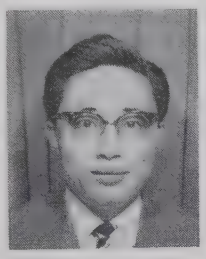
He joined the product development laboratory of the IBM Corporation, Poughkeepsie, N. Y., where he was engaged in the design of solid-state circuits. At present, he is studying full time at R.P.I. under an IBM Fellowship.

W. G. Strohm

Mr. Strohm is a member of Tau Beta Pi and Eta Kappa Nu.

❖

Frank F. Tsui was born in Tientsin, China, on May 4, 1922. He studied physics at the Yenching University, Peiping, China, where he received the B.Sc. degree in July, 1948. From 1949 to 1953, he did research on spark discharges at the Department of Electrical Engineering of the University of Liverpool, England, from which he received the Ph.D. degree in 1954.



F. F. Tsui

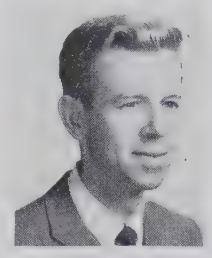
Since 1955, he has been a Research Fellow at the Institut fuer elektrische Nachrichtentechnik und Messtechnik of the Technische Hochschule of Munich, Germany, doing research work on transistor circuits and magnetic-core memory design.

Dr. Tsui is a member of Phi Tau Phi, the German Association of Electrical Engineers (VDE), and the German Communication Engineering Society (NTG).

❖

E. K. Van De Riet was born in Brandon, Wis., on June 18, 1924. He received the B.E.E. degree, specializing in communications, from the University of Minnesota Minneapolis, in 1948, and the M.S. degree in electrical engineering, specializing in computers, from the University of California, Berkeley, in 1953.

In 1953, while at the University of California, he supervised circuit and logical design work on CALDIC. From 1943 to 1945, he served in the U. S. Navy. From 1948 to 1951 he was a field engineer with the Schlumberger Well Surveying Corporation of Houston, Tex. From 1953 to 1955, he was a Junior Engineer, later a Senior Engineer, at Marchant Research, Inc., Oakland, Calif., doing small-scale digital computer work. In 1946 he was employed by Marchant Calculators, Inc., in Oakland, as a Senior Engineer supervising work on small transistorized digital computers.



E. K. Van De Riet

In 1958, he joined the staff of Stanford Research Institute, Menlo Park, Calif., where he has been working on multi-aperture magnetic devices for computers.

Mr. Van De Riet is a member of Eta Kappa Nu and Sigma Xi.

❖

Hannon S. Yourke (A'55–M'60) was born in New York, N. Y., on October 17, 1926. He received the B.S.E.E. and M.S.E.E. degrees from Massachusetts Institute of Technology, Cambridge, in 1954.

H. S. Yourke

From 1954 to 1955, he was employed by the Norden Laboratories of White Plains, N. Y., doing work on the development of microwave rotary joints. He joined the IBM Research Division, Poughkeepsie, N. Y., in 1955, and was a member of the advanced circuit development group until 1958. During this period, he invented the transistor current steering circuits and worked on the development of circuitry for the STRETCH Computer. Presently he is manager of the Exploratory Circuits Group of the IBM Advanced Systems Development Division, Peekskill, N. Y.

Reviews of Books and Papers in the Computer Field

E. J. McCluskey, Jr., reviews editor
T. C. Bartee, J. S. Bomba, W. J. Cadden, D. C. Engelbart, and M. Lewin, asst. reviews editors

Please address your comments and suggestions to the Reviews Editor, Professor E. J. McCluskey, Jr., Department of Electrical Engineering, Princeton University, Princeton, N. J.

A. Combinational Switching Circuit Theory And Boolean Algebra

R61-36 The Simplification of Multiple-Output Switching Networks Composed of Unilateral Devices—G. C. Vandling. (IRE Trans. on Electronic Computers, vol. EC-9, pp. 477–486; December, 1960.)

There are a vanishingly small number of problems in switching theory that are tractable enough to yield to a complete solution. The classic example of one so amenable is the minimization of a single switching function in a two-level (normal) form. The author has shown, to quote him, ". . . that two-level multiple-output switching networks composed of . . . diodes can be simplified or minimized in much the same manner as single-output networks." His paper proves this statement.

The method, in brief, is as follows. Let $F_1, F_2, \cdots, F_q$ be the given functions to be realized by a single network. Determine all the prime implicants of the products of the functions taken, $1, 2, \cdots, q$ at a time. Form a prime implicant table, with a row for each prime implicant so generated and a column for each minterm of each given function. Proceed in the standard manner of McCluskey,[1] determining essential prime implicants, deleting dominated rows and columns, and then selecting a minimal set of the remaining prime implicants. In this final selection, a weight is assigned to each prime implicant which is equal to the number of diodes required for its circuit realization. Don't cares are taken into account.

The paper is derived from a Master's thesis (Syracuse University) and as such, the material is quite detailed. Unfortunately, it is too detailed for the present publicaton. Material is offered that is of no apparent use; for example, two theorems are stated and proven but never again mentioned. Auxiliary variables, q in number, are introduced but are extraneous; the assignment of weights as mentioned is sufficient for minimization.

Since the method yields a minimum, it is noteworthy. However, the computations required are in most cases prohibitively lengthy. One of the author's examples involves three functions in three variables and yields 12 prime implicants and 11 minterms for the chart. The reviewer chose a 5-variable, 5-function example "at random" and came up with 214 prime implicants and 93 minterms; after removing essential prime implicants dominated rows, the chart had 65 prime implicants and 80 minterms. Clearly, the problem can get out of hand even for a relatively small number of functions.

The author's statement that "this algorithm can readily be programmed on a digitalcomputer" seems to be unjustified in some cases. The reviewer is of the opinion that a "true minimum" is not necessarily the golden ideal. Use of a computer in switching circuit design is sometimes too expensive when that is attempted. "Near-minima" are frequently preferable, particularly when one can save considerable time at little sacrifice in circuit cost. The 5-variable problem, for example, is particularly well suited to such a compromise approach.

P. M. Sherman
Bell Telephone Labs., Inc.
Murray Hill, N. J.

[1] E. J. McCluskey, "Minimization of Boolean functions," *Bell Sys. Tech. J.*, vol. 35, pp. 1417–1444; November, 1956.

R61-37 Switching Function Canonical Forms Based on Commutative and Associative Binary Operations—Peter Calingaert. (AIEE Paper 60-1199, AIEE Fall General Meeting, Chicago, Ill.; October, 1960.)

Promoting a new canonical form for switching circuits may be justified if any advantage over the present schemes can be gained. The connectives proposed for the "new canonical form" are neither new nor better. Those engaged in the design of switching circuits are inclined to accept as canonical the union of minterms and consider the conjunction of maxterms already as rather inconvenient.

Speaking from the viewpoint of logical designers, several points appear objectionable:

1) The combinations of connectives suitable to form a ring are well established and imply the equivalent abstract system of a Boolean algebra.[1] Symmetric difference and conjunction are a textbook example. Because no attractive switching elements are available to mechanize these connectives (or their complementary equivalence), there is little interest in using these connectives.
2) The reviewer disagrees with the statement that the "Lukasiewicz and the Sheffer functions respectively, are precisely the only two binary operations either of which alone suffices to represent an arbitrary switching function in terms of its argument." It can be shown that there is a much broader group of connectives which qualify as "universal" without being commutative, associative and/or distributive.[2]

Taking one of the asymmetric functions R of the three denoted by the author as $K=2, 4, 11, 13$ might suffice to demonstrate this point:

$$K = 13 \cdots . \qquad R(a, b) = a + b'. \tag{1}$$

A Boolean algebra is a lattice with 0 and 1, both being available in any digital circuit. Therefore, the following relation can always be implemented:

$$R(0, a) = 0 + a' = a'. \tag{2}$$

Similarly, the substitution below is always possible:

$$R(R(0, a), b) = a' + b'. \tag{3}$$

This relation is the Sheffer stroke obtained by iterations of an asymmetric function and will therefore satisfy the conditions of a "universal" function. A similar proof holds for the other three asymmetric functions of two variables.

In summary, this paper is a fine exercise in some basic concepts of algebra. It does not appear to be useful from an engineering point of view.

Ernest J. Schubert
Consultant
557 E. Walnut St.
Pasadena, Calif.

[1] N. Jacobson, "Lectures in Abstract Algebra," D. Van Nostrand Co., Inc., Princeton, N. J., vol. 1, p. 209; 1951.
[2] E. J. Schubert, "Universal modulus, matrix logic VI," *Commun. and Electronics*, no. 52 (*Trans. AIEE*, pt. 1, vol. 79), pp. 778–780; January, 1961.

R61-38 Computational Aids for Determining the Minimal Form of a Truth Function—Ronald Prather. (*J.A.C.M.*, vol. 7, pp. 299–310; October, 1960.)

The proposed algorithm finds the extremals of a function by examining each vertex of the "cares" to see whether it and all its adjacent vertices are contained in a single cocycle (prime implicant). If the extremals constitute a cover of the function the method is fairly efficient. The notation is similar to that used in PKMIN2,[1] an IBM 704 program distributed through SHARE.

The treatment is considerably less general than that already available. Multiple output minimization[2] is not mentioned. The "less-than" operation[3] of Roth for uncovering higher-order extremals is not mentioned. A procedure for picking a cover from the nonessential cocycles is not given; in fact, a misleading statement is made concerning picking a cover—"Preference is of course given to those cells that cover the largest number of vertices." This is true only if something comparable to Roth's "less-than" operation is used. Canonical form is convenient, but limits the set of feasible applications.

With some straightforward extensions, the notation and method could be combined with that of Roth producing a workable hand method.

William W. Boyle
IBM Corp.
Poughkeepsie, N. Y.

[1] A. E. Randlev, "Computation of a minimum two-level and-or switching circuit, PKMIN2," *SHARE*, April 22, 1959.
[2] D. E. Muller, "Applications of Boolean algebra to switching circuit design and error detection," IRE Trans. On Electronic Computers, vol. EC-3, pp. 6–12; September, 1954.
[3] J. P. Roth, "Algebraic topological methods in synthesis," *Proc. Internatl. Symp. on the Theory of Switching*, April 2–5, 1957, in "The Annals of the Computation Laboratory," Harvard University Press, Cambridge, Mass., vol. 29, pt. 1, pp. 57–7 3

R61-39 Principles of Mechanization of the Analysis of Circuits with Contact-Containing Relays—P. P. Parkhomenko. (Inst. of Automation and Remote Control of the Academy of Science, USSR. Presented by Academician V. S. Kulebakin, July 17, 1958.)

The author presents sets of equations which "form the basis of a special logical machine for the analysis of relay circuits."

For combinational circuits, the equation is simply the product of the sum of products canonical form of the function and the product of sums form for the inverse. This formulation permits taking into account both direct paths and shunt paths.

The author defines a "combinational equivalent" for sequential circuits which contains arguments for both primary and secondary relays. Output functions are included for each of the secondary relays as well as for the primary outputs. Using this combinational equivalent, he then obtains tables of combinations for varying times and initial conditions under the assumption that the times of operation and release of all elements are identical and equal to $\tau > 0$. This condition is later weakened to make the operation times of some of the secondary relays integral multiples of τ. He presents block diagrams of the relay circuit analyzer (which tell one virtually nothing of the construction) and makes claims regarding its capacity, usefulness for analyzing noncontact networks and for solving "logical and other problems connected with the design of relay devices."

A noteworthy omission is the lack of any discussion of the time required to perform the analyses.

Warren Semon
Sperry Rand Res. Ctr.
Sudbury, Mass.

R61-40 Statistical Estimation of Provability in Boolean Logic—Antonín Špaček. (*Trans. Second Prague Conf. on Information Theory, Statistical Decision Functions and Random Processes*, Liblice, near Prague, June 1–6, 1959, Publishing House of the Czech. Acad. Sci., Prague, Czech., pp. 609–626; 1960.)

This paper, written by a well-known probabilist, essentially commences with a familiar result of the modern theory of Boolean algebra to the effect that the collection of theorems, and *a fortiori*, the collection of tautologies of the propositional calculus P can each be interpreted as a finitely generated Boolean filter;[1,2] in fact, as a principal Boolean filter, since P can be formulated with a single axiom.[3] The author points out, without reference to the literature, the existence of a similar nonprobabilistic algebraic theory for 1) the first-order functional calculus with equality (see, *e.g.*,the abstract of Tarski and Thompson[4]); the reviewer points out the existence of such a nonprobabilistic theory for 2) the pure first-order functional calculus[5] and a more developed version of 1).[6] The author says that he will treat 1) in a future paper.

The author defines a "Boolean logic" as an ordered pair (A, I), where A is a Boolean algebra and I is a Boolean filter. Let $I(a)$ denote the principal filter in A generated by $a \in A$ [*i.e.*, $\{x: x \in A \text{ and } a \leq x\} = I(a)$, where the partial-ordering is the one induced by the lattice]. Then $(A, I(a))$ is said to be an "axiomatic theory" in A generated by a. If $x \in A$ is a well-formed formula (*wff*) of A and $x \leq a$ then x is said to be a "specialization" of the axiom a; (a is assumed to be a nonatom of the algebra). He introduces the concept of a "random sampling" in a Boolean logic as a certain sequence of specializations of a. This in turn is used to formalize the idea of a "heuristic proof," *viz.*, as a certain statistical estimate. Then to each *wff* of $(A, I(a))$ there is made to correspond a random variable, which is the length of a heuristic proof for determining whether or not the *wff* is a theorem of $(A, I(a))$.

He correctly shows, among other things: 1) that if a *wff* of $(A, I(a))$ is not a theorem of $(A, I(a))$ then, under certain conditions, the heuristic proof of its unprovability has finite length with measure 1 on a certain probability space;[7] and 2) that the length of the heuristic proof of unprovability for every *wff* of $(A, I(a))$ which is not a theorem of $(A, I(a))$ has finite moments of all orders.

The reviewer points out that methods similar to those of the paper under review have some immediate relevance to theories associated with the concepts of "definability"[8,9] and "heuristics."[10]

A. A. Mullin
University of Illinois
Urbana, Ill.

[1] R. Sikorski, "Boolean Algebras," Springer-Verlag, Berlin, Göttingen, and Heidelberg, Ger.; 1960, 176 pp.
[2] P. R. Halmos, "Boolean Algebras," Dept. of Math., Univ. of Chicago, Chicago, Ill.; 1959, 175 pp. (Unpublished.)
[3] A. Church, "Introduction to Mathematical Logic," Princeton Univ. Press, Princeton, N. J., vol. 1; 1956, 376 pp.
[4] A. Tarski and F. B. Thompson, "Some general properties of cylindric algebras," *Bull. Am. Math. Soc.*, vol. 58, abstr. no. 85, p. 65; January, 1952.
[5] P. R. Halmos, "Algebraic logic III," *Trans. Am. Math. Soc.*, vol. 83, pp. 430–470; November, 1956.
[6] P. R. Halmos, "Algebraic logic IV," *Trans. Am. Math. Soc.*, vol. 86, pp. 1–27; September, 1957.
[7] J. L. Doob, "Stochastic Processes," John Wiley and Sons, Inc., New York, N. Y.; 1953, 654 pp.
[8] J. Robinson, "Definability and decision problems in arithmetic," *J. Symbolic Logic*, vol. 14, pp. 98–114; June, 1949.
[9] K. L. De Bouvère, "A Method in Proofs of Undefinability with Applications to Functions in the Arithmetic of Natural Numbers," North-Holland Publ. Co., Amsterdam, The Netherlands; 1959, 64 pp.
[10] H. A. Simon, A. Newell, and J. C. Shaw, "Report on a general problem-solving program," *Proc. Internatl. Conf. on Information Processing*, Paris, France, June 15–20, 1959, Columbia Univ. Press, New York, N. Y.; 1960.

B. Sequential Switching Circuit Theory and Iterative Circuits

R61-41 Cycles in Logical Nets—John H. Holland (*J. Franklin Inst.*, vol. 270, pp. 202–226; September, 1960.)

This paper examines the relation between the complexity of the cycles in a logical net and the complexity of the resulting behavior. The author's method is to use periodic input as a tool for analyzing net properties. By this interesting procedure, he is able to prove some theorems that would be very awkward to prove in any other way.

A periodic input sequence to a net causes a periodic output sequence, and the latter is modified in certain general ways by the complexity of the cycles. Two questions are considered. First, what is the relation between the period of the input and the period of the output? The author answers this for nets containing simple cycles. Second, is there some level of complexity c, such that any behavior possible for any net can be realized by a net containing only cycles of complexity $c' \leq c$? For specific cases, this second question is answered in the negative; *i.e.*, certain behavior possible using cycles cannot be realized by using other cycles of lower complexity. Here, complexity does not mean the number of cycles or their size but only how many subcycles a cycle can have.

It might be thought that cycles serve just to provide nets with memories of various recycling times. In this case, by reducing complex cycles to simple cycles, all behavior possible by using complex cycles could be realized by using just simple cycles. But this is not the case, as the author clearly shows. Cycles of more than order 1 (simple cycles) are needed for certain behavior, *i.e.*, the full range of behavior cannot be obtained by using just simple cycles. This is an interesting and rather surprising result. For example, a corollary shows that no net of simple cycles can count to the base 3. Thus cycles do not function *just* for information storage.

The paper concludes with operations generating extension from simple cycles to the next level, and then to the next, etc., up to, or almost to, logical nets in general.

James T. Culbertson
Math. Dept.
California State Polytechnic College
San Luis Obispo, Calif.

R61-42 Automata and Finite Automata—C. Y. Lee. (*Bell Sys. Tech. J.*, vol. 39, pp. 1267–1295; September, 1960.)

More explicitly than ever before, the author treats finite automata as a subclass of Turing machines so that confusion concerning their relationship should disappear. Besides a familiar type of Turing machine called a T-machine, the author defines a programmed type of Turing machine called a W-machine (after Hao Wang, who first considered this type of machine). The W-machine has a tape, can print 1 or 0, can move left or right as can a T-machine; whereas the action the T-machine takes depends on the symbol on the scanned square and its "internal state," the action of the W-machine depends on a program. A program is a list of numbered (addressed) instructions among which are the printing and moving instructions as well as a "transfer to program address A if the square under scan contains 1, otherwise transfer to the next program address on the list." Two machines (one may be T- and the other W-) are said to be *completely equivalent* (ce) if for every initial tape configuration (a finite sequence of 1's and 0's) c_0, the determined sequence $c_0, c_1, c_2, \cdots, c_i, \cdots$ of tape configurations such that $c_i \neq c_{i+1}$ is the same in both cases. Theorem 1 states that for every W-machine with b instructions, there is a ce *T-machine* with not more than b states, and that for every T-machine with s states, there is a ce W-machine with not more than $10\,s+1$ instructions. Concerning the number of internal states of the T-machine as compared with the number of instructions of the W-machine, it should be noted that, as here formulated, the W-machine performs acts more primitive than the T-machine. Theorem 1 enables one to better understand the frequently observed phenomenon that the activity of constructing Turing machines is similar to the activity of programming digital computers. Those familiar with programming will find the author's construction of a Universal W-machine quite interesting, although verification of details may require a wide blackboard and a sturdy eraser.

A finite automaton is a T-machine whose "table of moves" does not include the act "move left." The author further requires that the symbol the finite automaton prints be independent of the symbol on the scanned square. Why the author makes this further requirement puzzles the reviewer, for 1) it is not essential, and 2) it causes "bumpiness," *e.g.*, W-machines which do not contain the instruction "move left" correspond via the author's conversion procedure to "finite automata" which fail to satisfy the further requirement and another step is then taken to correct this.

A partial finite automaton A does not necessarily have a "next state" for each input state (symbol on scanned square) and internal state. Associated with A is a set R_A of tape configurations (finite sequences of 0's and 1's) defined as follows. The sequence $i_0, i_1, i_2, \cdots, i_n$ is in R_A if there exists internal states $q_0, q_1, \cdots, q_{n+1}$ such that $f(i_k, q_k)$ is defined for each $k=0, 1, 2, \cdots, n$, where q_0 is the initial state of the automaton and f is the "next state" function. Two automata, A and B, are ce with respect to $S \subseteq R_A \cap R_B$ if they are ce in the usual sense except that initial tape configurations are restricted to being in S. The author's Theorem 3 states that A and B are ce with respect to $R_A \cap R_B$ if they are ce with respect to the set of sequences in $R_A \cap B_B$ of length at most $ab-2$ where $a, b>1$ are, respectively, the number of internal states of A, B. It should be noted that if $R_A = R_C$ is the set of all finite sequences of 0's and 1's (*i.e.*, A and C are complete) and R_B is empty, then A and B are ce w.r.t. $R_A \cap R_B$, and B and C are ce w.r.t. $R_B \cap R_C$; but A and C are not necessarily ce w.r.t. $R_A \cap R_C$. The significance of the theorem is this: if A is a partial automaton which expresses design requirements in the sense that one seeks a complete automaton B ce to A w.r.t. $R_A = R_A \cap R_B$, then Theorem 3 shows how to test whether a given complete automation satisfies the requirement. In particular, by enumerating complete automata with 1, 2, 3, etc., internal states, and testing each in turn for satisfaction of the requirement, one can find a minimal complete automaton satisfying the requirement. While this procedure is, in general, not practicable, it at least gives a first approximation to a more efficient algorithm. The author's Theorem 4 establishes that for every a, b there are partial automata A, B with number of states respectively a, b which are ce w.r.t. sequences in $R_A \cap R_B$ of length less than $ab - \min(a, b)$, but not w.r.t. longer sequences.

By using essentially the same argument as the author, J. D. Rutledge and the reviewer jointly have strengthened Lee's Theorem 3 to the following *theorem:* A and B are ce w.r.t. $R_A \cap R_B$ if they are ce w.r.t the set of sequences in $R_A \cap R_B$ of length at most $a_1b_1 + a_0b_0$, where a_i, $0 \leq a_i \leq a$, is the number of internal states of A which yield the output i (0 or 1). This theorem holds for any number of input states and has an obvious generalization to any (finite) number of output states. Moreover, by the author's examples for Theorem 4: for every a, b, there are automata A, B with numbers of internal states a, b, respectively, which are ce w.r.t. sequences in $R_A \cap R_B$ of length less than $a_1b_1 + a_0b_0$ but are not ce w.r.t. sequences in $R_A \cap R_B$ of length (at most) $a_1b_1 + a_0b_0$.

There are some innocuous inaccuracies. On page 1288 $X^* = \phi u \cdots$ should be $X^* = \{\phi\}u \cdots$. Where "$\{\phi\}$" denotes the unit set consisting of the null set . On page 1286, line 4, after "would be ab-min (a, b)," insert "if $m \leq n$." The two concepts of finite automaton indicated by the author, one as a special kind of T-machine and the other given by i, ii, iii, iv on page 1279, do not exactly coincide because of differences in the state-symbol pairs for which action is defined. The reader will be able to provide inessential changes which remedy this.

Calvin C. Elgot
IBM Res. Ctr.
Yorktown Heights, N. Y.

R61-43 Summer Institute for Symbolic Logic Summaries—Cornell University, 1957. (Communications Res. Div., Inst. for Defense Analyses, Princeton, N. J., 2nd ed.; July 25, 1960.)

This volume has 83 articles, 17 of which are in the fields of switching theory, theory of automata, computer design and computer programming, and will be reviewed here.

The longest paper in the volume is A. Church's "Application of Recursive Arithmetic to the Problem of Circuit Synthesis." It is a complete paper and not a mere summary, although Church refers to it himself as "notes." It is a revision made a year later of what was presented at Cornell.

The basic problem of the first part of Church's paper is that of synthesizing finite automata (or sequential machines) satisfying conditions described in a system of symbolic logic known as restricted recursive arithmetic. A finite automaton is given, also, as a particular set of formulas in this system (expressing the relationship among state variables, input variables and output variables). This method of description is then extended to growing automata.

Church's formulas are long and clumsy, and it is not unfair to say that this paper demonstrates a practical disadvantage in Church's method. The theory is interesting, however, and one can gain much from reading Church's paper without glancing at a single formula.

J. H. North, in "A Machine Evaluation of Logical Building Blocks," investigates the use of a multi-input device that can serve as a universal building block, which can realize various logical elements by having an appropriate set of their inputs biased in certain appropriate ways.

C. Katz gives a brief description of "The Math-Matic Programming System," and C. Yehling gives a description of the "Flow-Matic System." These rather short summaries prompt many unanswered questions, one of which is, why are straightforward reports on programming systems (not particularly suitable for programming logical problems) presented at an institute for symbolic logic?

In a short two-paragraph summary, J. B. Rosser ("The Relation Between Turing Machines and Computing Machines") takes the position that despite many technologically significant developments, modern computers do not constitute any conceptually significant change over Turing machines.

In "Remarks on Finite Automata," M. Rabin and D. Scott present part of the material presented in their paper[1] that has since appeared in print.

In "Multi-Valued Switching Functions," M. J. Gazalé (Ghazala) investigates p-valued switching circuits made up from various types of building blocks. A mathematical result is proved for the case in which the fundamental building blocks are those realizing sum and product modulo p.

B. Dunham in "Symbolic Logic and Computing Machinery, a Survey and Summary," briefly describes the effect machines have on logic, and vice versa. Machines are used to program and execute algorithms in the field of logic, as in any other field. Also, there is hope that machines will solve problems (in logic as in any other field) in which they will bring forth the method of solution (artificial intelligence). On the other hand, logic is helpful in its application to machines, in ways too familiar to readers of these TRANSACTIONS to be repeated here.

In "Symbolic Representations of Calculating Machines," H. Wang briefly investigates various languages of symbolic logic for describing finite automata. He also makes some broad comments on the role of logic in the machine art.

D. M. Brender discusses the use of an IBM 650 in programming "The Logical Procedures Needed for Finding the Minimals of a Boolean Function on a Digital Computer."

In "The Quine Algorithm,"[2] R. Fridshal points out that for some truth functions the number of prime implicants is so large that any method of minimizing a truth function requiring a complete list of prime implicants is unmanageable. Fridshal presents three alternatives to any such method:

"1. Limit the class of problems to be solved. 2. Discover a new approach which might not be so unwieldy. 3. Determine an approximate minimum and not necessarily an absolute minimum."

Unfortunately, he does not discuss these three alternatives further.

M. Davis, in "A Program for Presburger's Algorithm," discusses in detail a program written for the Institute for Advanced Study digital computer which carries through one of the fairly well-known decision procedures of symbolic logic. He includes a description of the system of symbolic logic being worked on, and a brief description of the computer. He concludes on the basis of experience programming Presburger's algorithm that "with equipment presently available, it will prove impracticable to code any decision procedures considerably more complicated than Presburger's," such as Tarski's procedure for elementary algebra.

M. Kochen, in "Reliability of Automata," discusses the general problem of how to construct a machine of specified reliability from parts that are less reliable.

H. Gelernter, in "Theorem Proving by Machine," discusses in general terms the program on the IBM 704 that proves theorems of Euclidean plane geometry. He describes briefly how the program works, and the general significance of the program.

In "Two Logical Minimization Problems," J. P. Roth first describes briefly how to minimize a normal form by his "topological" method. He then shows how to generalize this method to minimize a certain generalization of the normal form.

R. M. Friedberg, in "An Experiment in Mechanical Learning," discusses his program on the IBM 704 that has the machine learn to recognize similarities. If a randomly made choice leads to positive reinforcement, there is greater probability of making a similar choice in the future; and the computer will learn its own criterion for similarity.

ROBERT MCNAUGHTON
Moore School of Elec. Engrg.
University of Pennsylvania
Philadelphia, Pa.

[1] M. Rabin and D. Scott, "Finite automata and their decision problems," *IBM J. Res. & Dev.*, vol. 3, pp. 114–125; May, 1959.

[2] An expanded version of this paper appeared in the following reference: B. Dunham and R. Fridshal, "The problem of simplifying logical expressions," *J. Symbolic Logic*, vol. 24, pp. 17–19; March, 1959.

R61-44 Design of Combinational Switching Circuits Using an Iterative Configuration—D. L. Epley. (*Proc. 4th Midwest Symp. on Circuit Theory*, Marquette University, Milwaukee, Wis., December 1–12, 1959; pp. M-1–M-23.)

This paper consists for the most part of a well-written tutorial review of work done previously by McCluskey[1–3] on the design of iterative switching circuits. The paper defines such circuits to consist of cascades of N identical combinational logic cells, $C_1, C_2, \cdots, C_N$, where each C_i has an external input x_i; an external output z_i; a set of carry variable inputs coming from the left $y_1, y_2, \cdots, y_k$; and a set of carry variable outputs going to the right $Y_1, Y_2, \cdots, Y_k$. The cells are constructed from either relays or AND, OR, and NOT electronic gate elements.

Epley adopts McCluskey's general method for designing these circuits, which is based on the space-time analogy between sequential and iterative circuits. His design steps are, in order: 1) construction of a flow table for a typical cell from verbal specifications; 2) assignment of binary k-tuples to states in the flow table; 3) construction of a typical cell; and 4) removal of unnecessary circuitry at the beginning and end cells. The main part of the paper centers on steps 2) and 4). For step 2), in case relay elements are used, Epley suggests McCluskey's method.[3] When electronic gate elements are used, he proposes the use of a "state-transformation table" defined as follows: The column headings of the table consist of all coded left input states, $y_1, y_2, \cdots, y_k$, and the row headings all coded right output states, $Y_1, Y_2, \cdots, Y_k$. The table entries are blank except when the corresponding column input state can be mapped by some cellular x input to the corresponding row output state. In this case, the table entry is the x input value which effects the mapping. The value of such a table is that it furnishes a reasonably efficient way to select the $y_1, y_2, \cdots, y_k$ state variable code so that the amount of cell logic is reduced. No optimality claims are made, however. Epley's comments on step 4) are elementary but neat. Several appropriate examples are included.

The original contribution of the paper is the state-transformation table as an aid to the problem of selecting good state variable codes. That this is indeed an important and difficult problem has been amply demonstrated,[4] but Epley's results seem only incremental in that he does not structure the problem significantly. (The reviewer anticipates publication of some of J. Hartmanis' recent work which does accomplish some of this structuring.) Nevertheless, Epley is to be commended for an interesting and teachably clear treatment of the subject.

W. L. KILMER
Dept. of Elec. Engrg.
Montana State College
Bozeman, Mont

[1] E. J. McCluskey, Jr., "Iterative combinational switching networks—genera design considerations," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-7, pp. 285–291; December, 1958.

[2] E. J. McCluskey, Jr., "A comparison of sequential and iterative circuits," *Commun. and Electronics*, no. 46 (*Trans. AIEE*, pt. 1, vol. 73), pp. 1039–1044; January, 1960.

[3] E. J. McCluskey, Jr., "Assignment of carry variables in iterative circuits," *Commun. and Electronics*, no. 52 (*Trans. AIEE*, pt. 1, vol. 79), pp. 772–778; January, 1961.

[4] E. J. McCluskey, Jr. and S. H. Unger, "A note on the number of internal variable assignments for sequential switching circuits," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-8, pp. 439–440; December, 1959.

C. Pattern Recognition and Learning Theory

R61-45 Design for a Brain—W. Ross Ashby. (John Wiley and Sons, Inc., New York, N. Y.; 1960. 280 pp. and 4 index pp.+ix pp. $6.50.)

This second edition of a well-known book has been quite thoroughly rewritten in the later parts. However, the main theme remains the same—that a deterministic machine of a particular type can exhibit behavior like that shown by animals and human beings, and in particular, that goal-seeking, and adaptive or learning behavior, as well as evolutionary processes, can be simulated.

The particular "machine" chosen here to represent the environment and organism together was suggested by the concept of "homeostasis." It has state-determined trajectories forming "fields" containing stable points, and has the ability to change fields in "step" fashion when certain variables are exceeded, and search for a new field (adaptation) with a stable point within the chosen limits of the variables

(homeostasis). Dr. Ashby calls such a system "ultrastable." He has built a hardware prototype, the homeostat, which exhibits some of the main features of the ultrastable system.

In its simplest form, possessing little preassigned structure, the ultrastable system must search an impractically large set of fields before finding a suitable one, and by then it will have lost all trace of its previous adaptations. The later parts of the book are largely devoted to discussing ways of avoiding these difficulties. Mainly, these involve partitioning the machine so that the parts are relatively independent. This should allow adaptations to take place serially and independently, if the parts have the correct design.

Throughout most of the book (except for a short mathematical appendix), emphasis is on reasoning by analogy and plausibility arguments. Most of these arguments are reasonably convincing, except those dealing with the more complex systems toward the end, where the specialist would prefer more rigorous general arguments, or more definite and completely worked-out examples, since the systems discussed become complex indeed, and their exact behavior is by no means obvious. Of course, this complexity, which is necessary to produce the complex behavior required, is just what makes it difficult to prove rigorous theorems.

Nevertheless, it seems likely that the desired brainlike behavior can be imitated as closely as desired by some such system, if only because the model chosen has such great generality. However, this generality itself creates a difficulty, for the value of the model in the science of the brain will depend upon finding actual measurables which relate naturally to an "ultrastable" system. No such variables are described in the book, and, indeed, they could be very hard to find, since there remains the possibility that the ultrastable system is not a "natural" model. Thus, it must be said that "Design for a Brain" does not carry us very far toward an understanding of the brain itself, although it puts forward many very interesting possibilities. Specialists may also be disappointed that Dr. Ashby did not find it possible to include discussions of related work in specific directions published since the first edition.

As a result, the book is probably most useful to the nonspecialist or general reader who is interested in learning what complex behavior may be exhibited by a deterministic system. Many are still not aware that such a system has the possibility of imitating "intelligent," "adaptive," or evolutionary behavior, and to them, the many ingeniously chosen examples from biology and psychology illustrative of the correspondence between "brainlike" and "ultrastable" behavior, and the clearly written discussions, should prove fascinating as well as instructive.

Belmont G. Farley
M.I.T. Lincoln Lab.
Lexington, Mass.

R61-46 Redundancy Exploitation in the Computer Solution of Double-Crostics—Edwin S. Spiegelthal. (*Proc. Eastern Joint Computer Conf.*, New York, N. Y., pp. 39–56; December, 1960.)

Given a list of empty storage locations, we are asked to fill them with English words (taken from a fixed standard dictionary, say). Such a puzzle would hardly be interesting if there were no constraints on how the words were to be chosen for the various locations. In the (simplified) version of Double-Crostics considered by the author, the constraints specify: 1) the length of each word, 2) that certain letters of one word must match certain letters of other words, 3) that some locations can only be filled by words from given restrictive lists, and 4) that a subsequence of the words must form an English sentence. Constraint 4) is not actually employed in the machine program, but is envoked in demonstrating the success or failure of the method on particular examples. The reviewer would guess that all the examples tried had at most one solution. Hence, either the program finds the unique answer in a reasonable amount of time, or it gives up. The author does not seem interested in giving an algorithmic method that would produce exhaustive multiple solutions, but rather wants practical and efficient techniques. Of course, a direct exhaustive search through all possibilities is out of the question in this version of Double-Crostics, as it is in all interesting problems of this kind. The author's aim was to formulate some general principles of heuristic programming to be built into a control (or "policy-making" or "decision sequencer") portion of the routine to cut down the need for long searches. In this way, he hopes to exploit the "redundancy inherent in structured-data problems." The main aspects of the program specifically oriented to the English language were its concern with the frequency lists for words and letters and with good and bad two- and three-letter combinations. This last is clearly connected with constraint 2), which causes letters to pop up in many different places when one location is being filled. Though only a small amount of grammar is thus built in, the results were good in the examples described in the paper. The way in which the various combinations were actually scored by the policy maker was not entirely clear to the reviewer, but the author's remarks are suggestive and reasonable and can best be judged by reading the paper. It is hoped that the experience gained from Double-Crostics can be applied to questions of machine translation.

Dana Scott
University of California
Berkeley, Calif.

R61-47 Machine Recognition of Spoken Words—Richard Fatehchand. ("Advances in Computers," Franz Alt, Ed., Academic Press, New York, N. Y., vol. 1, p. 207; 1960.)

The author presents a well-balanced and accurate survey of work done and problems remaining in the field of mechanical speech recognition. The paper begins with a concise description of the speech process, goes on to list and discuss in some detail the elementary speech sounds of English, and concludes with an explanation of current speech recognition machines and a discussion of various schemes which have been proposed for inclusion in future machines. The list of references provides a good representative cross section of the literature in the field. The reviewer knows of no more satisfactory presentation of the current status of the mechanical speech recognition problem.

Almost all of the work which has been done in the field of mechanical speech recognition was carried out before large-scale digital computers were generally available. The author mentions briefly some early applications of computers to speech problems and argues that the use of large computers may permit the realization of some of the complex recognition schemes that he discusses. This paper is in no sense a progress report on the success of computers in the speech recognition field, but rather a statement of the problems involved and a vote of confidence for the computer as a powerful tool for their eventual solution.

J. W. Forgie
M.I.T. Lincoln Lab.
Lexington, Mass.

R61-48 A Method of Voice Communication with a Digital Computer—S. R. Petrick and H. M. Willett. (*Proc. Eastern Joint Computer Conf.*, New York, N. Y., pp. 11–17; December, 1960.)

This paper describes a project to recognize spoken words with the aid of a digital computer. The words to be recognized are processed by an 18-channel Vocoder, the output of which is time sampled to provide the digital input to the computer. After time normalization so that each word contains a fixed number of equally spaced patterns, a word to be recognized is compared to a set of masks prepared from examples of the recognition vocabulary spoken by (optimally) the same speaker. Word recognition approached in this way apparently yields to a fairly simple pattern recognition scheme. The criterion for matching is reported as the following:

$$C = \left(\sum_{i=1}^{N} X_i Y_i \right) \Big/ \sum_{i=1}^{N} (X_i - Y_i)^2$$

where X_i is the ith spectral component of (presumably) a sample from a spoken word and Y_i is the corresponding component from the mask. This criterion in conjunction with the method for computing the masks is characterized by the authors as one of many plausible alternatives. Whether it was derived empirically, theoretically, or by intuition is not mentioned.

The results obtained were quite good. When the masks were prepared from the speaker's voice, identifications were obtained with almost 100 per cent accuracy. However, when the masks were not prepared by the speaker, the recognition varied from nearly perfect to

consistently in error. An attempt was made to overcome this difficulty both by making composite masks from good speakers and writing a program designed to adapt the masks to a new speaker. These attempts met with essentially negative results. In general, the results reported by the authors would have been more interesting if they had discussed the variables relating to the production of the words such as instructions to speaker, speaker training if any, signal-to-noise ratio, etc.

The problem of recognizing a word from a limited vocabulary is much easier than the problem of recognizing the sequence of phonemes in an arbitrary utterance. However, the former problem lends itself to a pure pattern recognition approach, whereas the latter requires considerations involving acoustic phonetics and linguistics, disciplines which are neglected to the ultimate dismay of some speech recognition projects. It appears that the authors have leveled very big guns at the limited vocabulary problem (a digital computer and readjustment for individual speakers), but a stronger case can be made for using these expensive and cumbersome guns in attacking a phoneme recognition problem. The real challenge in the limited vocabulary recognition field lies in developing a device which will operate over a wide range of individuals without adjustment, preferably in an analog manner or with digital equipment of a more simple nature than a general-purpose computer.

As to the presentation of the paper, one would have appreciated a more specific account and additional details both to make more clear the operations performed and to give a better idea of what conclusions can be drawn as a result of what appears to be a worthwhile exercise.

J. E. Dammann
Data Systems Div.
IBM Corp.
Poughkeepsie, N. Y.

R61-49 An Adaptive Character Reader—P. Baran and G. Estrin. (1960 IRE WESCON Convention Record, pt. 4, pp. 29–41.)

This paper describes a character recognition method based upon maximizing the *a posterio i* probability of the input pattern being a certain character, given the quantized ink distribution. Binary matrices of size 30×32 were used to represent the input pattern. In order to use Bayes' equation for the computation of these probabilities, the authors made the familiar but invalid assumption of independency among matrix elements, but supported this assumption on the basis of machine simplicity. However, other assumptions which were made were not stated. For instance, it was assumed that the *a priori* probabilities of class occurrence were equal, and that equal sample sizes were taken to estimate the various probabilities. It also seems that the number of samples satisfying a certain outcome was used in the numerical work in place of the associated probabilities; otherwise, the trick of letting $\log 0 = 0$ would be invalid.

The experimental results were based on machine printed numbers covering a wide range of quality from the IBM 407 line printer. The data used in testing the method were different from the data used to determine the probabilities. The results are rather mediocre (91 per cent recognition), considering that machine printing was used. Furthermore, some very highly correlated noise was present in the form of a portion of the next number in the line of printing appearing in the field of view. Since the numbers were apparently printed in numerical order, all sevens had a part of an eight in the right-hand portion of the field, etc. No discussion of centering problems was given, but an apparently powerful method for rejection was made evident from the experimental results.

The discussion of the Sibling machine for the implementation of this method seems incomplete. In particular, one notes that white areas have a different set of weights associated with them than black areas; yet, only one photographic mask per character is indicated.

The authors include an interesting discussion of the application of their method to language translation in an effort to justify the experimentally determined error rates. This discussion is, in fact, of general interest to character recognition since it applies to any character reading system. However, in this development, certain inherent assumptions again were made without statement concerning various types of independencies of the errors made in reading a text.

There were some minor errors noted in the paper. In the discussion of the language processing machine, the authors refer to $R = Y/D$ as a probability; it is $1/R$ which is the probability. There are obvious typographical errors in (4) and (5), and in Fig. 11 the dotted line segment and the light solid line segment above the 1 on the abscissa should be interchanged.

The title of the paper contains the word "adaptive," which evidently refers to the process of estimating the probability of occurrence of marks in an element by observing a sequence of data. The reviewer wonders if such a grandiose term ought to be applied to the classical statistical process of estimation.

This paper has covered a class of pattern recognition methods which certainly ought to be studied. Except for some minor errors and a vagueness about certain points, the authors have succeeded in investigating these methods rather thoroughly, but seem to be somewhat optimistic about their results.

W. H. Highleyman
Bell Telephone Labs., Inc.
Murray Hill, N. J.

R61-50 Filter—A Topological Pattern-Separation Computer Program—Daphne J. Innes. (*Proc. Eastern Joint Computer Conf.*, New York, N. Y., pp. 25–37; December, 1960.)

This paper introduces relatively new application of a data processing system to computer literature. The intepretation of the information on photographs of nuclear tracks in a bubble chamber is one example of applications where the input to a computer comes from a photograph. In this case, about a million photographs are produced each year at each bubble chamber installation. These photographs must be interpreted with "subcritical" manpower expenditures

The interpretation of bubble chamber photographs involves the following steps:

1) The recognition of nuclear events of interest on the photographs.
2) The measurement of the position of points on possibly pertinent tracks of an event on two or more views of each event.
3) The spatial reconstruction of points along each track.
4) The correction of these points to account for physical factors and the fitting of the corrected points with an appropriate space curve.
5) A kinematical analysis of each event.

Filter is an IBM 704 or 709 program for performing the third of the above steps for a specific class of nuclear events called star-type events. The input to Filter is data from a mechanical photographic film scanner (called the Spiral Reader) scanning film known to contain star-type events.

Each bubble chamber photograph contains 10^8–10^9 resolvable picture elements. Since all tracks and other marks on a photograph are contained in only a small fraction of these elements, it is more efficient to encode sample coordinate positions of all marks on each film than to encode and process a bit image of each photograph. The Spiral Reader generates a magnetic tape record of the polar coordinate position with respect to a position near the center of the nuclear event of all marks oriented in a generally radial direction. Filter separates nuclear tracks connected with the star-type event from other marks or tracks by accepting only track coordinate positions which lie along a continuous curve from the center of the event.

L. A. Kamentsky
IBM Res. Ctr.
Yorktown Heights, N. Y.

D. Digital Computer Systems

R61-51 High-Speed Counter Requiring No Carry Propagation—W. N. Carroll. (*IBM J. Res. Dev.*, vol. 4, pp. 423–425; October, 1960.)

This is an interesting circuit and it is clearly described. It is surprising that Mr. Carroll does not mention other efforts to solve the problem and give a comparison. Richards[1] mentions two which are considerably simpler. He does not cite references, but they surely are more than five years old. These depend on the principle that adding

[1] R. K. Richards, "Arithmetic Operations in Digital Computers," D. Van Nostrand Co., Inc., Princeton, N. J.; 1955.

binary one to a register will change the state of a flip-flop if all lower order flip-flops are reading one at the start of the process.

Fig. 1 shows a simple block diagram for using this principle to speed up the carry operation. The pulse is directly applied to the flip-flop of zero order and to gate G_4. If this flip-flop is found to contain a one, gate G_4 is enabled to pass the pulse to the first-order flip-flop and to gate G_5. The small delay D is needed to ensure that the gate cannot be enabled by a transition from zero to one in time to pass an improper trigger pulse to the next stage. In some designs, the flip-flop itself might serve this function, or a small RC delay would serve. This circuit shortens the carry time from the sum of the transition times of the registers to the sum of the propagation times of the gates in cascade. This can make a large improvement. The gates are simple two-input AND gates.

Fig. 2 shows a somewhat more complex circuit which can produce simultaneous changes in all affected stages of the register if suitable account is taken of wiring distances and a dummy gate introduced in the input to the zero-order stage. The pulse to be counted is applied in parallel to AND gates such as G_4, G_5, and G_6. G_4 is always enabled and may be omitted if small delay is insignificant. G_5 is enabled if the zero-order stage contains a 1; G_6 is enabled if both lower-order stages contain ones. Delays D serve to prevent false operation due to change of state occurring while the pulse is present. For a long register these gates become complex. Richards[1] points out that a marked reduction of gate complexity in a long-register may be attained at the cost of a modest carry time such as that of two gates in cascade, by combining the methods of Figs. 1 and 2. Comparing Fig. 2 with Carroll's Fig. 1 for equal length of count, we find the same number of binary cells, the AND gates such as G_4, G_5 and G_6 are more complex by one input than gates OR_1 and OR_2, but no equivalents of gates G_1 through G_7 are required. Furthermore, a true binary number is presented after each count, so readout complexity is reduced by more than 50 per cent.

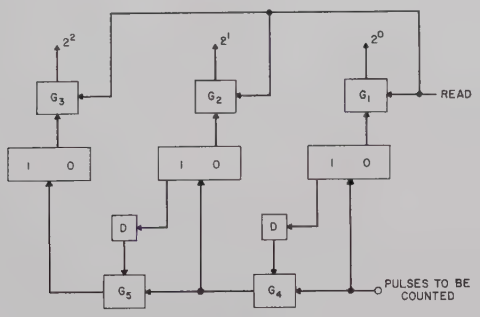


Fig. 1.

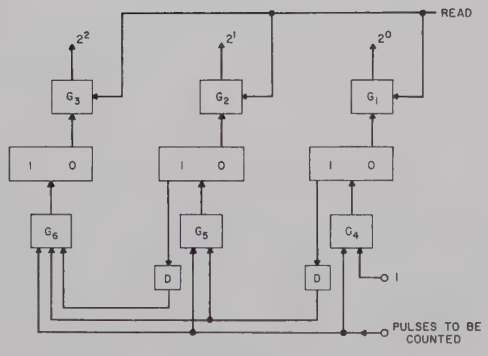


Fig. 2.

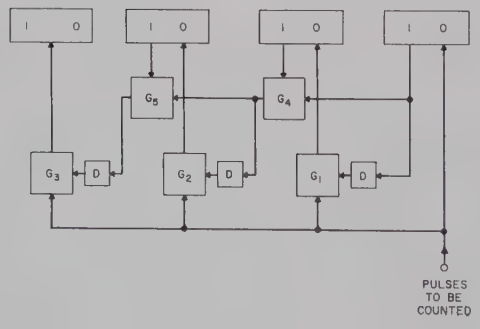


Fig. 3.

Fig. 3 shows still another way of carrying out the logical process and setting all stages at one time. This arrangement uses nearly twice as many gates as does Fig. 2, but all are simple two-input gates. The pulse to be counted is applied to the zero-order flip-flop and to gates such as G_1, G_2, and G_3 in parallel. Gate G_1 is enabled by a one stored in the zero-order flip-flop. Gate G_2 is enabled from G_4 if both zero- and first-order flip-flops contain ones. Each transmission gate in turn is enabled if all lower-order stages contain ones. The constraint on speed and length of the register is that the propagation time of gates such as G_4 and G_5 in cascade must be less than one counting interval so that all transmission gates may be enabled by the time the succeeding pulse arrives. Delay circuits D may be needed in low-order stages as in Fig. 2, but probably are not needed for higher-order stages because the cascade gates would provide some delay. Blair[2] has described a reversible binary counter employing the logic of Fig. 3. He claims simultaneous changes of all stages to the new state.

In conclusion, Mr. Carroll has given a clear description of a workable high-speed counter which appears to have no possible "race" conditions that might necessitate delays such as D but it works at the price of presenting the accumulated count and its complement in alternation. This greatly complicates the readout circuitry. It appears that other counters exist which can work as fast and always present the true binary number.

J. O. Edson
Bell Telephone Labs., Inc.
Murray Hill, N. J.

[2] R. R. Blair, U. S. Patent No. 2,954,485; filed December 24, 1956, issued September 27, 1960.

R61-52 Some Reflections on Digital Computer Design—W. Renwick. (*J. Brit. IRE*, vol. 20, pp. 563–572; August, 1960.)

The author's reflection on digital computer design touched upon a variety of topics dear to the heart of many a conscientious computer engineer. These topics include:

1) Reliability at the component, circuit, and system level; and the preventive measures for improving the effective reliability.
2) Component tolerances and their distributions; and the choice of "worst case" and "statistical" designs.
3) Relative merits and limitations of serial and parallel operations, synchronous and nonsynchronous operations, dc and pulse coupling; and the implications on components and circuits.
4) The "basic building block" arrangement; the optimum size and types of building blocks; interconnection and fabrication techniques.

Also mentioned in the paper are the newer components and techniques, including *p-n-p-n* devices, tunnel effect devices, magnetic logic circuits, cryogenic devices, phase-locked oscillators, and majority logic.

The paper is tutorial in nature and appears in the form of general comments. It is understandable that the wide coverage of subjects in a paper of this length allows only brief mention of each topic. Some of the problems mentioned in the paper were not adequately defined or illustrated; and the new components and techniques were presented without qualifying comments on their merits and limitations. Nevertheless, this paper serves its purpose well in providing a bird's-eye view of the tasks encountering the modern computer engineers.

Arthur W. Lo
IBM Corp.
Poughkeepsie, N. Y.

R61-53 The Impact of Automation on Digital Computer Design—W. A. Hannig and T. L. Mayes. (*Proc. Eastern Joint Computer Conf.*, New York, N. Y., pp. 211–232; December, 1960.)

In recent years, it has become commonplace for computer development organizations to employ present generation computers in "bootstrap" type of operations which aid in the development of future generation data processing equipment. Where large digital systems are being built, it is in fact almost an economic necessity to use some form of machine-aided design. The reporting of efforts in this field, however, has to a large extent been confined to reporting accomplishment rather than techniques and methods. This paper is no exception.

From the title of this paper, a more philosophical work was expected—one which perhaps discussed the relationship of automation in the factory to the design task. There is instead reported a rather impressive automatic design process which begins after the basic creative effort of the engineer is finished and ends with the production of manufacturing information and other design documentation. A set of computer programs is employed to assist in arranging and interpreting Boolean expressions which describe the logic of a new machine; to assign logic elements to plug-in circuit cards; and finally to provide wiring information for interconnecting circuit cards. Logic schematics showing logic-flow, together with equipment terminal information, can also be produced by a computer. The reader may find the language of this paper somewhat esoteric; but at least with careful reading he will learn the kind of design tasks a computer can perform when properly programmed.

It is probably not controversial to say that programming computers to perform design tasks characterized as being routine and repetitive (and not very creative) is, at this stage, an art mostly devoid of the scientific approach. Possibly this is because, up until now, programs written have been closely allied to hardware design, and the techniques employed have not been generalized to an extent that make them applicable to other users. Until they are so generalized, papers like this one serve mainly the purpose of keeping workers in the field abreast of what is going on about them.

G. A. Sellers, Jr.
Bell Telephone Labs., Inc
Whippany, N. J.

R61-54 Use of a Digital Analog Arithmetic Unit Within a Digital Computer—Donald Wortzman. (*Proc. Eastern Joint Computer Conf.*, New York, N. Y., pp. 269–282; December, 1960.)

In the creative stage of any new technology, there is inevitably an efflorescence of novel suggestions. Experience has taught us that this public airing of ideas (which some of them need very badly) is an indispensable adjunct to progress. Even a marginal idea can often suggest a sound one, and occasionally an ugly duckling becomes a swan. This appears to be the stage that combined analog-digital computation is painfully passing through today, and the literature abounds with imaginative schemes[1–3] such as the one presented in this paper. Such speculation is healthy and useful and, hopefully, will ultimately converge on the practical hybrid computers of the future.

The author suggests a configuration of digital-to-analog converters, summing amplifiers, analog multiplexers (gates), comparators, and digital control logic peripheral to a standard digital computer. The interconnection of these peripheral units at any given time is determined by the multiplexers activated under the control of the digital computer's stored program. In addition, data words from the digital computer can be stored in the digital register associated with each D/A converter. Conversely, analog results can be encoded and transferred to the digital computer.

The basic computing element is the digital-to-analog converter, whose output is the analog product of the digital input and the analog reference input. The over-all Digital/Analog Arithmetic Unit employs 18 such D/A converters (presumably each with a 17-or-more bit storage register), 24 operational amplifiers, one variable gain scaling amplifier, 76 analog multiplexing gates (each under the control of the digital computer), one comparator and an unspecified quantity of digital logic for control of signs, encoding, transfers, timing, and automatic scaling. In the background lurks a general-purpose digital computer of unspecified capabilities which presides over this melange. The logical mechanism by which this control is exercised is not treated.

Two examples are given by the author of typical applications; one, an eight-term polynomial expansion for e^x and the second, the multiplication of two eight-by-eight matrices.

In trying to capitalize on the high effective computing speeds obtainable by using an analog or analog-digital configuration set up to carry out several steps simultaneously, the author is on sound ground. The trick is to achieve this saving efficiently from the viewpoint of cost and equipment complexity.

Some of the author's conclusions regarding achievable accuracies are questionable. In tests on single conversion sequences (D/A followed by A/D), the author measured errors of $\pm$ 0.05 per cent of full scale. With this experimental data in hand, he then assumes that high-speed D/A conversions would be possible with "average" accuracies of 0.001 per cent and A/D conversions with "average" accuracies of 0.01 per cent. Both of these specifications exceed the generally acknowledged state of the art. The author then postulates that his arithmetic unit consisting of 16 D/A converters, 24 amplifiers, and 72 gates, also could have an accuracy of 0.001 per cent, as if component errors don't sometimes add as well as cancel.

The author is also somewhat misleading on his speed estimates. He implies, for example, that the computation of a single element in the matrix problem could be accomplished in 100 μsec, but this does not include the time required to store the elements of the multiplying matrix in the D/A input registers. Indeed, it would appear that most of the computation time consumed in the use of the Digital/Analog Arithmetic Unit would be in transferring data from the digital computer to the various D/A converter registers. During such transfers, the digital computer would not be available for other operations.

The author's schematic of the D/A converter-tester combination, which was the only equipment actually built and tested, implies that the matching of two D/A converters and not absolute accuracy was measured. Aside from this detail, the author's presentation of his interesting concept was clearly stated.

Mark E. Connelly
MIT Electronic Systems Lab.
Cambridge, Mass.

R61-55 The Instruction Unit of the IBM Stretch Computer—R. T. Blosk. (*Proc. Eastern Joint Computer Conf.*, New York, N. Y., pp. 299–324; December, 1960.)

In the design of Stretch, some extremely novel and hitherto untested ideas are incorporated in a computer having massive complexity in its controls and data paths. The objectives are speed at arithmetic, an elaborate interrupt mechanism, and a wide-ranging order code with 12 distinct instruction formats. Concurrent operation of different parts of the machine are the means to the end. This paper does not provide sufficient information on the performance of the instruction unit (I unit) as part of the system. Hence, the reviewer feels extremely frustrated because he is unable to evaluate the achievement which is potentially a very important one.

The I unit, including interrupt, contains over 53,000 transistors—the largest single unit in the machine. 60 per cent of these transistors are for control and decode logic, and the paper gives a detailed explanation of the operations performed. All instructions are fetched from memory by the I unit, and are first error-checked. Then the I unit performs any indexing operations required, initiates core memory read-outs, and loads suitably modified instructions and other information into a device called lookahead. Lookahead serves as a buffer between the memory and the arithmetic unit. When the contents of an index register are changed, its old value is loaded into lookahead. At a later time, the old value may be recovered and returned to the index register if an interrupt occurred or if a branch order was obeyed. The I unit always assumes that arithmetic conditional branches will not be obeyed, and prepares further instructions in sequence. A large number of operations are done concurrently by the I unit. In particular, the prefetching and error-checking of further instructions, and the buffering of information on the way to look-

[1] *Proc. Combined Analog Digital Computer Systems Symp.*, Philadelphia, Pa.; December 16–17, 1960.
[2] H. K. Skramstad, "A combined analog-digital differential analyzer," *Proc. Eastern Joint Computer Conf.*, Boston, Mass., pp. 94–100; December, 1959.
[3] R. C. Lee and F. B. Cox, "A high-speed analog-digital computer for simulation, IRE Trans. on Electronic Computers, vol. EC-8, pp. 186–196; June, 1959.

ahead are paralleled with the preparation of the current instruction. Hopefully, lookahead and the arithmetic unit are also busy during this period.

The paper states that the high performance goals for this device were achieved. It does not state what these goals are or why speed is important for the I unit. It seems to the reviewer that the I unit is the most critical area in the machine in terms of speed required, for the following reasons:

1) All instructions go to the I unit, while only some instructions go to the arithmetic unit. Time spent in completely executing instructions such as "count-and-branch" is charged against the I unit only.
2) In the nature of things, the I unit is frequently idle because a memory is busy or lookahead is full. In its busy periods, it must keep up the correct average rate of speed.
3) Instructions have relatively different execution times in the I unit and in the arithmetic unit. Some take a long time in one unit and a short time in the other. This implies a higher execution rate to look after peak load problems.
4) Some I unit operations are later invalidated by subsequent interrupt or the occurence of an unexpected arithmetic condition. This time, and the time for restoring index registers to their old values, is strictly wasted, from the viewpoint of the I unit.

The reviewer has difficulty in judging the merits of the I unit because the degree of overlap between its operations and those of the arithmetic unit is not described. He hopes that the following correctly summarizes the information on performance given in the paper. The following times are the minimum times for the I unit to process an instruction:

1.2 μsec for an indexed floating point instruction
4.8 μsec for a count and branch instruction
3 to 6 μsec for a variable field length instruction.

No figures for duty cycle or average execution time are given, or for the duty cycle of the arithmetic unit.

The use of superlatives is unfortunate, especially in places where factual information would be more informative. After any computer is finished, its designers know of things that they would have preferred to do differently. Since none of these is mentioned, it is to be hoped that they will be discussed in a later paper by the same author.

D. B. Gillies
University of Illinois
Urbana, Ill.

R61-56 The Harvest System—P. S. Herwitz and J. H. Pomerene. (*Proc. Western Joint Computer Conf.*, San Francisco, Calif., pp. 23–32; May, 1960.)

This paper describes one of the first serious realizations of a processing unit designed exclusively for large-scale data processing problems. As such, the paper deserves serious attention. Unfortunately, adequate information is not always provided in this brief review paper.

As the authors observe,

> "Most of the information generated and handled in our society is either non-numeric or developed within areas supported by little or no theoretical structure. A general-purpose system for processing this type of information must be organized very differently from a more conventional scientific computer."

Although recognizing this, the authors have not had complete freedom to design a general data processing machine. Rather, their task was much more restricted: given the existing scientific computer STRETCH,[1] what additions to it will permit large-scale data processing? In particular, what processing unit (comparable to an arithmetic unit) should be appendaged to the STRETCH system (without altering its normal input-output and its very fine memory) to optimize computing efficiency where ordering, classification and statistical estimation of large masses of loosely organized data are the predominate problems.

[1] E. Block, "Engineering design of the STRETCH computer," *Proc. Eastern Joint Computer Conf.*, Boston, Mass., pp. 48–58; December, 1959.

The setting of the design problem biased its solution in two important respects. 1) In data processing, unlike scientific computation, the data may have internal symmetries that are largely destroyed by storage in a memory whose long, lean words were designed for scientific computation. For example, it is doubtful that machine recognition of two-dimensional geometrical patterns will ever be elegantly resolved by a processor whose memory destroys the X-Y translation and reflection symmetries of the input data. Indexing is not the sole answer to memory structure. 2) Unlike scientific computation, input-output can play a significant role in processing itself, as for example in the encoding that occurs directly at the retina, or in the improved data rate of a "character-at-a-time" as opposed to a "point-at-a-time" display CRT. Data processing for the Harvest system is largely an outgrowth of the data processing of conventional core stores, magnetic tapes and disk files. It may well prove unsatisfactory for, let us say, the automatic scanning of satellite photographs.

The standardized unit of information of the data processor of the Harvest system is the "byte": any consecutive 8 bits of 1 (or 2 consecutive) 64-bit words of the STRETCH memory. The organization of the data processor is reminiscent of a three-address serial computer (with the byte replacing the bit) in that two serial "streams" of bytes can be extracted from memory under independent indexing control, processed, and a single "sink stream" returned to memory under control of yet a third indexing unit. Direction of the data streams through the individual processing modules in many ways reminds one of the analog computer linkages, but here plugboard flexibility has been achieved at fast logic speed by setup instructions of 192 bits each (3 STRETCH words of 64 bits). No adequate description of the actual setup instructions is given in this paper.

Bytes are selected from memory according to a preasigned indexing pattern to keep a steady flow of selected bytes flowing through the processing chain and back into memory. Quoted maximum data rate is 4 bytes/μsec. For memory access, each stream has independent indexing of the initial address, increment and count type. At completion of the count, automatic referral to the next in a chain of indexing instructions is provided for. Regarding this data pickup the authors assert, "Particular attention has been given to direct implementation of triangular matrix selection and to the iterative chain of any formal inductive process, however complex." The burden of proof traditionally falls on the authors; no explanation is provided. I trust they will clarify this and many other clouded points in subsequent papers.

An example of interstream processing is comparison logic ($<$, $\leq$, $>$, $\geq$, $=$, $\neq$) between the bit P_i of the P source stream and the bit Q_i of the Q source stream. This permits search operations "abstracting from a master file all records whose control fields bear any specific one of six possible relationships to the control field of each record of a detail file." Merging and collating operations can be instrumented by control signals generated from the comparison operation.

Perhaps the most fundamental data processing operation, beautifully exploited by the Harvest processor, is table lookup and its decision logic ramifications. Here again is the evolution of a familiar programming trick into a wired-in family of instructions.

> "The table lookup facility consists of two units. The more important logically is the Table Address Assembler which accepts bytes from one or two sources and from them forms the lookup addresses which are sent to memory. . . . The other is the Table Extract Unit which permits selection of a particular field within the looked-up word. . . . The table lookup facility may be used to associate statistical weights with the occurrence of particular sets of bytes. For example, the occurrence of a byte P_i in the P stream together with a byte Q_j in the Q stream may be assigned a weight W_{ij}, which would be stored in a table and referenced by an address formed from both P_i and Q_j . . .
>
> A Statistical Accumulator (SACC) is provided . . . to sum the weights W over a succession of sets of bytes."

Three basic table lookup instructions (with variants) are provided. The first, a generalization of multiple deferred addressing, is the Sequential Table Lookup instruction:

> "a series of table references are made, each successive reference after the first being made in a table whose address is extracted automatically from the previously referenced table entry. Also, as each reference is completed, a variable amount of data may be adjusted according to the contents of the table entry (similar to the operation of the Turing machine)."

The second table lookup instruction is the (Indirect) Load-Store instruction: the byte of one of the two source streams is used in the Table Address Assembler to generate a sequence of primary addresses which become respectively "the origins of a field of data to be entered via the other source stream unit or the location at which the data field is to be stored by the sink stream unit; the data is moved from source to sink," the next primary address picked up and the cycle repeated. Indirect addressing is also printed.

The third table lookup instruction permits marking of the single bit in memory at the address computed by the Table Address Assembler. In the core memory this "marking" is restricted to ORing a one into the referenced bit position, but for the high-speed (1024-word) memories, "a one may be either ORed or added into the referenced bit position. . . . The ability to add ones into high-speed memory words permits use of these words as individual counters."

The data processor of the Harvest system is an excellent mutation in the evolution of data processing computers away from the traditional scientific computer—a split dating from the parallel processing by small (9-bit) characters in the Lincoln Laboratory TX-2 computer.[2]

We look forward to seeing results of detailed operational experience with the Harvest data processor as that experience accumulates. Points of interest will be the frequency of use of various instructions and chains of instructions, duty cycles of each processing box (comparison logic, table lookup facilities, marker comparison, etc.), running times for a well-defined class of problems, and recommended extensions or restrictions of the en route sensing and branching facilities. Such information is indispensable to the development and design of the next generation data processing computer.

B. H. McCormick
Digital Computer Lab.
University of Illinois
Urbana, Ill.

[2] J. M. Frankovich and H. P. Peterson, "A functional description of the Lincoln TX-2 computer," *Proc. Western Joint Computer Conf.*, Los Angeles, Calif., pp. 146–155; February, 1957.

R61-57 The RCA 601 System Design—A. T. Ling and K. Kozarsky. (*Proc. Eastern Joint Computer Conf.*, New York, N. Y., pp. 173–177; December, 1960.)

The RCA 601 computing system is described in this brief but very readable paper. The 601 has been designed from a very general system point of view to cover the broadest range of computer applications. It is all solid-state, modular, and asynchronous, with core memory and a general input-output control which may attach all the usual card, printer, paper and magnetic tape equipment and special-purpose devices, with simultaneity of up to 16 of these units with main frame processing. Machine instructions are built in the computer from sequences of "elementary" operations, and hence are easily changed and adapted to special situations. Examples of asynchronous machine operation are given.

Much attention has been given to the design of the programming aspects of the computer. The core memory is directly addressable by word and half word, and characters of several lengths are addressable *within* a half word (this is not the same as bit addressability of data without regard to word boundaries). The instructions are of variable length, with up to four addresses, and the address modification and indirect addressing abilities are unusually versatile. For example, arithmetic instructions are usually three-address. The three-address registers which hold the three addresses are automatically decremented by one after the instruction has been executed, and if any of these new values are appropriate for the data addresses of the next instruction, these addresses may be omitted in the coming instruction, and the appropriate identification bits included in the operation (half word) part. One half word is saved per missing ("assumed") address, decreasing both time and space required for the program. Operands found in the accumulator are handled similarly.

Some important features in the RCA 601 system are not mentioned, among these the limit register, the input-output queue table, multiprogramming features and the wide range of data formats and associated instructions (as three levels of precision of both binary and decimal arithmetic). The maximum size of the core memory is 32,768 56-bit words, and there is no auxiliary memory except magnetic tapes, which may prove to be a disadvantage of the system.

The three examples in Fig. 4 assume that the accumulator is initially 0, 0, and 1, respectively, and that the result is left in the accumulator. In Fig. 5 the first and second columns seem to have been erroneously transposed. According to the RCA 601 manual,[1] bits 1, 2, and 3 should designate the address modifier and bit 4 is the signal to increment address modifier. There are also several mistakes in the list of direct addresses in Fig. 6. Under the conditions stated there are only four possible direct addresses:

$$\begin{aligned} &1)\ C + (AM_1) \\ &2)\ C + (AM_1) + (AM_2) \\ \text{or}\quad &1)\ D + (AM_3) \\ &2)\ D. \end{aligned}$$

Neither a complete nor very detailed picture of the RCA 601 could be given in a few pages. The paper provides a well-written general description of the system's main elements, which are illustrative of the philosophy of versatility, generality and care that has motivated its design. Those who desire specific knowledge of the system's capabilities should get more particular information.

Sidney Fernbach
Lawrence Radiation Lab.
Livermore, Calif.

[1] "RCA 601 General Information Manual," RCA, Cherry Hill, N. J., 325 pp.; January, 1961.

R61-58 Computer Engineering—S. A. Lebedev, Ed.; transl. from the Russian by Jane Stuart. (English ed. published by Pergamon Press, New York and London, 1960. 184 pp.)

This little book is a collection of eight papers, many of them by members of the Institute for Exact Mechanics and Computing Technique, one of the institutes under the Soviet Academy of Sciences. Academician Lebedev, the editor of the volume, is director of the Institute.

The title "Computer Engineering" is misleading, since the book does not attempt to be an integrated treatise on this subject. Although some of the eight papers are concerned with engineering topics, the topics and the treatments are such as to provide very little new information or guidance for either the experienced computer engineer or the newcomer to the field. The titles of the papers are:

1) The Power Supply System of BESM.
2) Digital Integrating Machines (Differential Analyzers).
3) Dynamic Flip-flops and their Use in Parallel Computers.
4) A Method for Automatic Monitoring of a Serial Arithmetic Unit.
5) Methods of Selecting the Required Word from a Dictionary.
6) The Role of the Ferrite Core in a Matrix Storage Unit.
7) Reliability of a Matrix Type Magnetic Store with Linear Selection.
8) Basic Nomenclature and Definitions in Automatic Digital Computer Engineering.

The longest paper (74 pages) is the one on the relatively unimportant topic of digital differential analyzers. The paper is largely a rewrite of well-known information, 21 of the 22 listed references being from American publications. The paper on the BESM power supply discusses reliability of the tubes in the computer, the effects of heater voltage changes, and the stabilization of the power supply. Although these matters might have been of interest when the BESM was built some five years ago, they are not pertinent in these days of transistorized computers. The same may also be said for the paper on dynamic flip-flops.

The author of the short paper on checking an arithmetic unit almost discovers the rules for parity checks on arithmetic operations which were stated by Garner.[1] Another short paper, the one on dictionary search, is simply a restatement of the obvious. The two papers on core memories are readable discussions of the subject, but will convey little new information to anyone who has done even a little study in this area.

[1] H. L. Garner, "Generalized parity checking," IRE Trans. on Electronic Computers, vol. EC-7, pp. 207–213; September, 1958.

The last paper attempts to define a number of terms relating to computers in an attempt to standardize their use in Soviet computer technology. However, the translation from Russian to English has been so complete that the original Russian words are presented here in transliterated form rather than in Cyrillic. This is surely no service to the reader who might otherwise find such a list of nomenclature a valuable assistance in his own reading of original Russian literature.

The effort devoted to the publication of this book in English must lead us to ask whether more discrimination should not be used in the selection of Soviet material for translation. This particular book presents no new or useful information, and while its translation might be justified by virtue of the view it presents of Soviet computer technology, that view is not broad enough to be valuable or dependable.

Norman R. Scott
Dept. of Elec. Engrg.
University of Michigan
Ann Arbor, Mich.

R61-59 Digital Models—A. V. Shileiko. (*Automation and Remote Control*, vol. 20, pp. 1638–1687; December, 1959. English transl. published June, 1960.)

This paper is a survey in which digital differential analyzers, rate multipliers and related devices are described; all of these are classed as "digital models" because they "can be used for simulation in the true time scale of definite physical processes and, consequently, can operate as elements of automatic control systems."

Initially the descriptions are made from an essentially mathematical point of view, and are to the point and free of obscuring complications. A "digital model" is really a device which processes trains of pulses; the method of representing variables is called "delta-modulation," by which is meant the familiar pulse-per-quantum-step of a variable. The author readily passes from the initial general approach to specialized counters and rate multipliers, and then to the conventional DDA integrator arrangement. Here the functioning of an integrator, the pertinent mathematical relationships, the basic scaling equation and servo operation are all disposed of within three pages. The question of errors is mentioned briefly, and Shileiko shows awareness of the phase problems which can arise in the binary (vs ternary) incremental system.

The pass band of a low-frequency filter equivalent to an integrator is derived by considering the accuracy of an integrator and the iterations per second $1/\delta_t$ of that integrator. The accuracy is defined only to the extent of considering some error ϵ, greater than the quantization error. If the gain of an integrator is defined as $1/\epsilon$, then $1/2\pi \cdot 1/\delta_t$ is the gain bandwidth product of that integrator.

There is a table listing various DDA's, including machines from Australia, Japan and England, but none from the USSR. At the end of the paper there is an argument for classing digital models with analog machines, rather than with digital machines.

Shileiko's paper has value as a tutorial for those with some knowledge of DDA's and as an indicator of Russian interest in the subject.

Ramon L. Alonso
Instrumentation Lab.
M.I.T.
Cambridge, Mass.

E. Circuits and Components

R61-60 Wave Generation and Shaping—Leonard Strauss. (McGraw-Hill Elec. and Electronic Engrg. Ser., McGraw-Hill Book Co., Inc., New York, N. Y. 509 pp. +10 index pp. +xvi pp., $12.50.)

The aspects of electronic circuit theory which form the basis for treating pulse and digital circuits in a unified manner have received relatively little attention in the literature in contrast to the wealth of detailed design, analysis and performance characteristics presented on the basis of individual circuits. The need for a new approach to the whole subject is met most satisfactorily by Professor Strauss in this book through the use of piecewise linear circuit techniques as an analytical tool and through concentration on the mode of operation of active elements rather than on specific devices and their characteristics. While the functional aspects of the class of circuits involved is emphasized both in the well-chosen title and in the text, the book is concerned with the principles of circuit design. As is to be expected from the integrated nature of the approach adopted, it must be read as a whole to obtain a good understanding of the methods which it presents and it is not, nor is it claimed to be, a catalog of circuit information.

In keeping with the spirit of the title and of the circuit treatment adopted, the book is divided into five sections which proceed from an initial exposition of piecewise linear circuit methods to discussions of shaping, timing, switching, memory and oscillations. The first section covers both the development of piecewise linear models to represent the nonlinear characteristics of diodes, triodes, pentodes and transistors and the application of break point analysis to circuits for wave shaping. A point worthy of special mention is the complete integration of transistors and other semiconductor devices into the presentation at all stages of the text. The next section covers the principles of sweep circuits and the operating characteristics which can be realized using gas tubes, vacuum tubes and transistors. Switching is the topic of the third section, which is devoted to multivibrators and blocking oscillators and includes a chapter on negative resistance switching circuits. The fourth section has memory as its title, but is confined to a brief and introductory review of magnetic and dielectric devices as switching and memory elements. The concluding section deals with oscillations of the almost sinusoidal variety and with negative resistance oscillators.

The book is the fourth to appear in the Brooklyn Polytechnic Institute series and is based on class notes developed there over the past few years. The text material has been carefully prepared and has clearly benefited from classroom experience. Its value is further enhanced for both students and teachers through the inclusion of worked examples and of nearly 350 problems pertaining to all aspects of the subject matter.

The least satisfactory section is that dealing with memory, where the treatment is substantially below the over-all level of the book and has the appearance of having been included primarily to claim some coverage of this important topic. At other points, the discussion falls short of being complete, as in the development of the triode tube piecewise linear model which lacks a plate-cathode diode to take account of the saturation effect. The location of the chapter on negative resistance switching after two devoted to the multivibrator but before treatment of the blocking oscillator tends to obscure the role of this concept and its relationship to the regenerative treatment of switching. This defect is compounded by the relegation of a full discussion of negative resistance to the end of the last chapter of the entire book and, in consequence, phase plane locus methods are not used to the full extent possible. This appears to reflect a reluctance on the part of the author to use locus of operation diagrams to illustrate the role of active devices in the circuits considered. In this connection, it is to be noted that the entire treatment of waveshaping is carried through in terms of transfer characteristics and without any operating locus diagrams other than one devoted to simple ac and dc load lines.

Taken as a whole, this book is an important and useful contribution to the literature of pulse and digital circuits, and Professor Strauss is to be commended for developing and illustrating underlying circuit concepts in a logical and readable fashion. It is designed primarily for students in their senior year or in the initial stages of graduate study and merits serious consideration by instructors in search of a text for either one- or two-semester courses where the emphasis is to be on unifying ideas. This reviewer has no hesitation in recommending this book to students, teachers and circuit engineers who wish to learn and then exploit in actual circuit design the approach and method which its author advocates.

W. D. Jackson
Dept. of Elec. Engrg.
Res. Lab. of Electronics
M.I.T.
Cambridge, Mass.

R61-61 Esaki Diode Logic Circuits—G. W. Neff, S. A. Butler, and D. L. Critchlow. (IRE Trans. on Electronic Computers, vol. EC-9, pp. 423–429; December, 1960.)

The well-selected list of annotated tunnel-diode circuits published in this paper illustrates various powerful design concepts. As good introductory reading material should do, this well-written paper emphasizes circuit operation rather than intricate design considerations. The future of the tunnel diode probably rests in a new

genus of circuits that has been introduced with the invention of the bilateral active diode device. The paper reviewed discusses a few such circuits, two of which deserve further comment.

Tunnel diodes in combination with square-loop magnetic cores result in an unprecedented class of valuable logic circuits. In these circuits, high-speed logic functions can be coupled to information-storage capacity requiring no holding power. An original magnetic-core binary flip-flop published in the paper uses a core both as gate—to control the polarity of the trigger pulse applied to a bistably biased diode—and as memory. Upon initial application of the power supply current, the flip-flop always assumes one particular state. A worthwhile modification of this circuit should reliably store the core state even in case of temporary failures of either diode or power source. Because of the speed limitations of magnetic square-loop cores, the recovery time of a one-diode flip-flop published elsewhere[1] can be made smaller than that of a magnetic-core circuit. However, because of the wide current ratings of tunnel diodes, it seems likely that magnetic films can eventually be substituted for cores to speed up this magnetic core circuit.

Two-phase driver and temporary inductor storages of a tunnel-diode stepping switch (shift register) which has been explained in the paper, can be replaced by transistors as coupling elements between register stages. A common-emitter transistor coupling stage has sufficient response delay to transfer the contents from one stepping switch stage to the subsequent stage, it provides current gain useful for driving auxiliary circuits, and it introduces the polarity inversion needed to operate the shift register on a single square-wave pulse sequence.

The authors have succeeded in giving a good cross section of current tunnel-diode circuit practice. The paper is a good background for appreciation of the rapidly growing art in tunnel-diode logic circuits.

R. A. Kaenel
Bell Telephone Labs., Inc.
Murray Hill, N. J.

[1] R. A. Kaenel, "One-tunnel-diode flip-flop," Proc. IRE (Correspondence), p. 622; March, 1961.

R61-62 Calculated Waveforms for Tunnel Diode Locked Pair—H. R. Kaupp and D. R. Crosby. (Proc. IRE, vol. 49, pp. 146–154; January, 1961.)

Computer simulation has become a popular concept in current technology. An increasing number of papers are published yearly that report on the use of high-speed digital computers for circuit studies. The authors of the reviewed paper have resorted to simulation in order to control spurious impedances which are especially noticeable at high frequencies typical of tunnel-diode switching performance. In particular, the paper publishes output waveforms of tunnel-diode locked pairs that are driven by various sine wave voltages of different frequencies.

The danger of computer misuse is a serious hazard in modern engineering. More than ever, the question as to the warranted application of general-purpose computers deserves some thought, for clever programming routines might easily be over accentuated at the cost of candid problem analysis. The computer then becomes an end where it should be a means.

Computer misuse is principally connected with processing of *implicit equation systems*. In this category fall computer simulation of experiments. This type of computer application is certainly warranted if the experiments are easier, cheaper or faster to simulate than to actually set up and conduct. Simulation might then be used to test closed form approximations useful for predictions, to perform parameter fitting, or to optimize construction principles, and it might be an aid in defining and understanding the problem involved.

A computer used for the numerical evaluation of an *explicit closed form expression* takes the place of a slide rule or some other arithmetical aid such as a nomogram. It should be borne in mind, however, that closed form solutions are valuable to a greater audience than a set of numerical results, even if those solutions are much less accurate than these results. The generality of abstract solutions is never available in the specific numerical results of the machine. A large number of calculations are necessary to convey the same amount of information that is encoded in the well-studied and tabulated functions found in most mathematical handbooks.

The authors of this paper have gone through a certainly exemplary piece of numerical evaluation of a set of differential equations, but they have neglected to process the important results. Without excessive effort, I am sure that a simple and yet useful mathematical model could have been constructed that sufficiently accurately describes the well-known behavior of a tunnel-diode locked-pair circuit. Even though the problem must remain nonlinear, the circuit could then have been analyzed in a rigorous fashion and results would have been obtained which preserve all the important parameters of the transfer relationship. For instance, two parabolas have been used by this reviewer as an approximation to the tunnel-diode characteristic. The analytical results obtained by this approximation were derived in less time than would have been required by computer simulation. That the published "results agree with unpublished experimental data" proves that the stray impedances of the authors' experimental setup have no bearing on the circuit performance. This is certainly a worthwhile result—for the authors—but perhaps less so for the reader.

R. A. Kaenel
Bell Telephone Labs., Inc.
Murray Hill, N. J.

R61-63 Tunnel Diode Logic Circuits—R. H. Bergman. (IRE Trans. on Electronic Computers, vol. EC-9, pp. 430–438; December, 1960.)

Monostable type of logical tunnel-diode circuits have two interesting features. First, they do not require a reset pulse. When recovery speed is not at a premium, this may represent a significant saving compared with bistable-type circuits which absolutely need a reset pulse or, instead, a pulsating power supply. Secondly, each diode stage can be driven by a pure voltage source without suffering a loss of logical gain. Power dissipation in load resistors is thus avoided—or at least reduced—and, because of little heat generation, monostable-type circuits can be packed very densely.

The serious drawback of monostable-type circuits is their speed limitation due to the recovery process. However, since there are no additional recovery problems involved in substituting a transformer for the timing inductor, inversion finds a simple implementation in this type of circuit.

Yet, pulse transformer design for high-speed monostable circuits is difficult and costly. A closer look at the actual effect of these transformers could yield some very interesting results. This is certainly a worthwhile direction in which more work should be done.

The monostable tunnel-diode circuits presented in this paper are worth further consideration. It would have been helpful if the fairly complete analysis of the basic building block had been complemented by a comparison with alternate circuits that do not employ the monostable principle. The more difficult design of systems using monostable-type circuits might well be offset by more economical solutions.

R. A. Kaenel
Bell Telephone Labs., Inc.
Murray Hill, N. J.

R61-64 A Novel Adder-Subtractor Circuit Utilizing Tunnel Diodes—R. A. Kaenel. (1960 IRE WESCON Convention Record, pt. 3, pp. 53–64.)

This paper presents two counter configurations which exploit the properties of a tunnel diode flip-flop. The basic flip-flop configuration has been described before,[1] but this is the only detailed analysis of the circuit that has been published. It warrents careful study by anyone seriously interested in tunnel diode circuits; unfortunately it will require careful study to follow the analysis.

The paper is of interest for two reasons: 1) the circuit offers distinct advantages over other counters and it can be designed—some-

[1] W. F. Chow, "Tunnel Diode Digital Cicruitry," *Digest of Tech. Papers of the 1960 Internatl. Solid State Circuits Conf.*, Philadelphia, Pa., February 10–12; pp. 32–33. Also IRE Trans. on Electronic Computers, vol. EC-9, pp. 295–301; September, 1960.

thing which cannot be said of most published tunnel diode circuits, and 2) the method of analysis is applicable to any tunnel diode circuit.

The author spends the first quarter of the paper on a general introduction to tunnel diodes and counters. The fractional nanosecond switching speed of the tunnel diode is discussed as well as the limitation of the speed of light, the need for microminiaturization, and the desirably low impedance level of the tunnel diode. While interesting, it seems somewhat irrelevant when compared to the 1-Mc inductively coupled counter which is finally evolved.

The basic circuit makes use of the fact that two tunnel diodes in series yield a composite characteristic which has a positive resistance region bounded at both ends by negative resistance regions. By the appropriate choice of bias voltage and impedance level, positive or negative pulses can be selectively amplified or attenuated. Furthermore, the circuit can be designed to combine gate and memory functions, while maintaining sufficient power gain to drive another stage without intervening amplifiers.

The paper contains a good analysis of the basic flip-flop, including both static and dynamic load-line and diode characteristics. The significant parameters are given and typical values are stated. The expression for the recovery time after firing has a typographical error and should read:

$$\tau_r' = 2L\left[\frac{\Delta j}{\Delta u} - \frac{\Delta j}{\Delta u}\,\frac{(V_e - V_0)}{(V_e - V_s)}\ln\frac{(V_e - V_0)}{(V_s - V_0)}\right].$$

A major irritation in this paper is the author's loose terminology where specific numbers would be more appropriate. The junction capacitance of the diodes must be surmised from the statement that they were 20-ma "medium quality germanium diodes." The actual speeds of various transients are described as occurring with "practically no delay" or "virtually simultaneously." The actual switching delay is never stated.

In conclusion, the author states that the experimental results show that the circuits "have been proved experimentally feasible and and operative." A more detailed investigation of design tolerances remains to be done.

F. K. Buelow
IBM Corp.
Poughkeepsie, N. Y.

R 61-65

1) MAD-Resistance Type Magnetic Shift Registers—David R. Bennion. (*Proc. 5th Annual Conf. on Non-Linear Magnetics and Magnetic Amplifiers*, Philadelphia, Pa., October 26–28, 1960; pp. 96–112.)
2) Analysis of MAD-R Shift Register and Driver—David Nitzan. (*Proc. 5th Annual Conf. on Non-Linear Magnetics and Magnetic Amplifiers* Philadelphia, Pa., October 26–28, 1960; pp. 113–133.)

These two papers expand the growing literature concerning diodeless magnetic logic circuits utilizing transfluxors, *i.e.*, devices of the multiflux path type. Elimination of semiconductor elements permits interstage voltage levels for small cores in the millivolt range, and this, coupled with the lack of need for providing standby power, leads to very small power requirements. Also, since only bulk property magnetic and conductive materials are used and fabrication is simple, ruggedness and reliability can be obtained potentially at low cost.

In the first paper the author starts by comparing the two basic methods used thus far to insure flux conservation in these devices. In one approach (Prywes[1] and the MAD of Crane[2]), information-advancing pulses of limited amplitude are used; in the second, advancing pulses of nearly unlimited amplitude are applied interleaved with longer priming pulses of limited amplitude (Gianola[3] and Briggs and Lo[4]). Although the former approach gives faster and more efficient operation, the gain and operating margins are restricted by the limited drive, so that the latter approach is finding more practical application. A two-advance-pulse, single-prime-pulse version of this approach is described which the author calls a MAD-Resistance type because of the coupling loop resistances which are required to limit loop current flow during the priming phases. This circuit utilizes holding currents to prevent spurious information kickback during advance phases. The speed is moderate, in optimum form only four times slower than the MAD. The efficiency is especially poor as a result of the increased advance drive needed to overcome the kickback losses, but this is not an important factor in practical applications utilizing small elements.

An analysis of the MAD-R circuit is presented which seems disjointed, but the results obtained are summarized in an excellent table. Bennion's schematic circuit and flux path diagrams are hard to follow; the notation characterized by Fig. 1-B of Nitzan's paper, in which the cores are shown as they actually appear physically, with single-turn windings having immediately evident polarities, is much preferred by this reviewer.

Bennion next carries out an analysis to determine range boundary values for a typical circuit. In this analysis he has taken a new approach in using the rate of rise of the advance current, rather than the advance current itself, as one of the boundary parameters. This is done because ramp-shaped advance pulses yield better margins than rectangular advance pulses; he points out that elastic flux effects are lessened with such pulses. However, he does not state the equally important reason that such pulses also give more nearly rectangular coupling loop shift currents which result in improved transfer efficiency. He shows a calculated boundary diagram for the circuit considered, but makes no attempt to compare this with experimental observations because of approximations made in the analysis, the most serious of which is the assumption, standard thus far in the literature, of rectangular core output waveforms. He has also chosen for his circuit a less-than-optimum priming winding turns ratio which slows the circuit 25 per cent; the reasons for doing this are not clearly stated and need elaboration. The main reason for using this winding turns ratio appears to be to reduce the ratio of the magnetic field on the undesired priming flux path about the main aperture to the field on the desired path about the output aperture. For a material of perfect hysteresis-loop rectangularity, this latter ratio could be only slightly less than 1, but in practice, lower values are required for materials having less-than-ideal properties, to prevent partial priming flux change on the undesired path. For the optimum priming situation which yields fastest operation, the field ratio is 2/3, whereas it is 4/11 in the circuit analyzed by Bennion. Further study of the maximum value permitted for this ratio for different material compositions and core geometries would be desirable.

The paper by Nitzan presents, first, a more thorough analysis of Bennion's circuit in an attempt to set the problem up in a form suitable for computer solving and optimizing. He shows an equivalent circuit for each multi-aperture core and for the entire shift register and presents a matrix for facilitating the calculations. Unfortunately, the same rectangular waveform assumption must be made (reflected in the use of a constant switching resistance). Also, a constant switching inductance has been assumed which is even more untenable in practice. The author realizes the nonlinear nature of the problem and makes no attempt to carry out the calculations and to compare them with practical circuits in this paper, but points out that the approach taken may be useful as a future starting point.

The author presents next a discussion of the important concept of interstage flux gain, and relates this to the range map boundary problem in a clear and easily understood manner.

Finally, in what is perhaps the most valuable section of Nitzan's paper, experimentally determined waveform, flux gain, range boundary and operating speed data are presented for a representative circuit. This material is well organized and gives a clear picture of typical circuit operation.

George R. Briggs
RCA Labs.
Princeton, N. J.

R61-66 Self-Propagating Core Logic—A. S. Myers, Jr. (*Proc. 5th Annual Conf. on Non-linear Magnetics and Magnetic Amplifiers*, Philadelphia, Pa., October 26–28, 1960; pp. 74–95.)

This paper is a cursory discussion of a transistor-magnetic core logic mechanization scheme. The first section, titled the "Basic Circuit," establishes the notation convention, the three periods of operation—quiescent state, triggering and core switching, core resetting—

[1] N. S. Prywes, "Diodeless magnetic shift registers utilizing transfluxors," IRE Trans. on Electronic Computers, vol. EC-7, pp. 316–324; December, 1958.
[2] H. D. Crane, "A high-speed logic system using magnetic elements and connecting wire only," Proc. IRE, vol. 47, pp. 63–73; January, 1959.
[3] U. F. Gianola, "Integrated magnetic circuits for synchronous sequential logic machines," *Bell Sys. Tech. J.*, vol. 39, pp. 295–332; March, 1960.
[4] G. R. Briggs and A. W. Lo, U. S. Patent No. 2,968,795; filed May 1, 1957, issued January 17, 1961.

and the mechanization of what resembles a blocking oscillator in operation. In addition, the fundamental AND and OR functions are described. The author then briefly gives a word description and diagram of each of the following circuits: retiming or synchronizing, binary counter, decimal counter, and core-transistor trigger. The shift register is presented as an example of system application.

Two features of this paper are worthy of particular note. First, the author presents a "component count" comparison between this logic and the competing NOR transistor logic. Second, the real problems of system testing and power requirement of magnetic devices were honestly admitted.

The following quote from the author points out the salient features of this logic system.

"A result of the systems studies along with a comparison of the circuits and components for SPCL circuits and transistor logic circuits indicates some advantages and disadvantages for each type. The SPCL circuits have the *desirable characteristics:*

1. Complex logical functions are performed with only one stage of delay.
2. Fewer components are used that have aging effects.
3. Transistors and diodes with relaxed specifications can be used in the circuits.
4. Each stage is a pulse generator and thus the waveform of a pulse passing through a number of stages will not have its shape deteriorated.
5. Each logic circuit is capable of storing information for as long as desired.

A number of existing *disadvantages* are:

1. Existing SPCL circuits use approximately five times the power required for transistor logic circuits on a stage-by-stage basis. However, this ratio is lower on a systems basis.
2. Testing a pulse system is more difficult than a direct current system.
3. Toroidal core winders capable of economically winding many multiple turn windings on small cores are not yet available."

One should especially note Advantage 3 and Disadvantages 1 and 3. Also, one should be aware that a more detailed description of the principal circuit and its operating mode was presented by Guterman and Carey.[1]

Although this paper may have usefulness in a tutorial course in digital technology, this reviewer feels that in general the practicing engineer is far too burdened to peruse it.

R. E. Gould
Aeronutronic Div.
Ford Motor Co.
Newport Beach, Calif.

[1] S. S. Guterman and W. M. Carey, Jr., "A transistor-magnetic core circuit; a new device applied to digital computing techniques," 1955 IRE Convention Record, pt. 4, pp. 84–94.

R61-67 Switching Circuits Using Bi-Directional Non-Linear Impedances—T. B. Tomlinson, (*J. Brit. IRE*, vol. 19, pp. 571–591; September, 1959.)

The author presents some interesting circuit concepts dealing with the development of direct coupled logic circuits using bidirectional (symmetrical) nonlinear impedances. Two types of symmetrical nonlinear impedances are considered; these are:

1. "Constant voltage (CV) elements" (elements which approach zero resistance at finite voltage).
2. "Constant current (CI) elements" (elements which approach infinite resistance at finite current).

The most attractive circuit presented using CV elements is what the author chooses to call a "two decision" AND gate. This circuit provides two outputs, one of which provides the AND operation on the input variables and the other provides the OR operation (positive logic). Unfortunately, the signals provided at the two outputs are not centered about the same voltage reference.

Using CI elements, a circuit technique is presented which would enable the construction of simple majority logic gates if appropriate CI elements were available (no simple elements exhibiting the required characteristics are presently available). This technique is significant in the mind of this reviewer in that it presents the possibility of performing majority logic with dc signals without requiring signal levels which are plus or minus with reference to the voltage to which the load is returned.

Experimental work is reported on dealing with the use of silicon carbide disks to construct multi-electrode CV elements. It is concluded that "present day material is not sufficiently uniform to make multi-electrode devices a practical proposition."

Appreciation of this paper is a bit hampered by a certain amount of circuitous reasoning and a few questionable conclusions. The first half of this paper is devoted to a review of diode-transistor logic circuitry, which is included in order to form a basis of comparison with circuits using CV elements. In this portion of the paper, a discussion (including calculations) of the diode AND-OR sequence is presented (positive logic). The conclusion is reached that "To avoid multiplicity of design, it is common practice to restrict the number of OR gates to one only. Any additional OR gates are preceded by redundant AND gates thus permitting the design of a single 'package' in the sequence, AND-OR-AMPLIFIER." However, the discussion presented does not indicate that multiplicity of design is necessary, and an examination of the circuit requirements indicates that the circuit could be designed to drive any number of OR circuits up to a specified maximum. Although this reviewer does acknowledge that restricting the number of OR gates to only one may be desirable on the basis of other considerations than those presented, the conclusion that this procedure is necessary is particularly unfortunate in that it is later used as a basis of comparison with the CV AND gate where the author points to the ability of the nonlinear AND gate to drive more than one diode OR gate as an advantage.

Also included in the review section of the paper is a comparison of the drive requirements of the diode OR-AND-AMPLIFIER sequence vs the AND-OR-AMPLIFIER sequence (positive logic-*p-n-p* transistors). The author demonstrates that the OR-AND-AMPLIFIER sequence is considerably more efficient from the point of view of drive requirements and encourages its use as a basic building block. He later develops the NONLINEAR AND-DIODE OR-AMPLIFIER sequence using CV elements (this turns out to be the most practical configuration), and he compares the drive requirements of this configuration (which are quite severe) with the less efficient of the two diode techniques. It is my impression that a more meaningful comparison would be with the more efficient diode technique, considering that in a large system there would be little basis on which to choose between the two from a logic viewpoint.

In his discussion of the CV AND gate, the author points out that by the addition of a second diode and bias supply, double selection properties may be obtained. He shows an example where the second output is obtained from a three-input AND gate. Rather than stating that this additional output provides the OR operation (positive logic), he says "This provides a second output which falls from 0.4 E to zero when all inputs fall to zero. This is equivalent to a second 3-AND gate preceded by three inverter stages." These statements are not consistent using either positive or negative logic.

Despite a few difficulties of the kind described above, the nonlinear circuit techniques presented are novel and readers interested in new logic circuit techniques will find the paper interesting.

Hannon S. Yourke
IBM Corp.
Peekskill, N. Y.

R61-68 Domain Behavior in Thin Magnetic Films—Joseph W. Hart. (*Proc. 5th Annual Conf. on Non-linear Magnetics and Magnetic Amplifiers*, Philadelphia, Pa., October 26–28, 1960; pp. 15–21.)

This paper describes domain behavior in 2000 A permalloy films as observed by means of the Kerr magneto-optic effect. It is oriented toward possible application to nondestructive memory elements, although it is also of interest for domain-stepping logic applications. The geometry considered is a film that is placed close to 10-mil flat conductors, one for producing a magnetic field transverse to the easy direction and two parallel ones, placed 40 mils apart, for producing a field parallel to the easy axis.

The section on theory, which is mostly qualitative, is primarily concerned with the effect that demagnetizing fields have on the resultant domain structure. The most pertinent phenomenon discussed in the paper is the creation of an "irreversible" domain with the aid of a demagnetizing field. Consider a film uniformly magnetized along

the positive easy axis. A current is driven in the easy direction through a conductor running the length of the film. This current produces a transverse field greater than H_k in the strip of film under the conductor and less than H_k elsewhere in the film. The magnetization in this strip will rotate to approximately 90° from the easy direction while the rest of the film is relatively undisturbed. The demagnetizing field produced by the "magnetic poles" at the edge of the undisturbed part of the film is in a direction to force the magnetization in the strip past 90°. Then, when the transverse field is removed, the magnetization in the strip will rotate to the negative easy axis, leaving a strip domain of reversed magnetization. Whenever transverse fields are subsequently applied and removed, the same demagnetizing field will force the magnetization in this strip to fall back to the negative easy axis. In this sense, then, the strip domain is "irreversible." Since the equilibrium polarity of this strip domain depends upon the polarity of the rest of the film element, reversal of the magnetization in the rest of the film element would result in reversing the equilibrium state of this "irreversible domain."

The author also discusses how the superposition of an applied field in the plus (minus) easy direction, from two parallel conductors above the film, can produce a plus (minus) residual state in this strip, in the "controlled area" near the conductor cross-over points, regardless of the state of the rest of the film.

The experimental section of the paper includes Kerr magneto-optic photographs for several series of experiments that clearly demonstrate the domain behavior discussed. One series of photographs shows the process of obtaining a strip domain of reversed magnetization disrupted by a normal region in the controlled area; another series shows the process of obtaining a strip of reversed magnetization with a bulge in the controlled area.

The author's description of domain geometry and demagnetizing field distribution lacks clarity. For example, he describes the demagnetizing field as originating from the dipoles within a domain. The demagnetizing field distribution could be better visualized and understood if considered as originating from "magnetic poles," where the magnetic pole density is equal to the divergence of the magnetization. The demagnetizing field is then given by the gradient of the scalar magnetic potential produced by these "magnetic poles." It follows that the macroscopic discontinuities in the component of magnetization normal to domain walls and film edges determine the demagnetizing fields.

The primary value of the paper lies in the experimental results illustrating some of the domain properties of thin magnetic films.

Victor W. Hesterman
Computer Techniques Lab.
Stanford Res. Inst.
Menlo Park, Calif.

R61-69 A New Magnetic High-Speed Switching Element—Its Application to Machine Tool Numerical Positioning Control—M. Dumaire. (*Proc. 5th Annual Conf. on Non-linear Magnetics and Magnetic Amplifiers*, Philadelphia, Pa., October 26–28, 1960; pp. 54–67.)

Square-loop magnetic core logic circuits have grown in large numbers since the Wang and Woo static magnetic shift registers and the Ramey half-cycle magnetic amplifier of approximately 10 years ago. This paper describes another core-logic circuit which superficially differs little from other developments in this area; however, it does have some distinctions that are worthy of note.

The Symmag element is of the two-core-per-bit, two-phase, set-reset family used as an inverting or complementing switch. The inverted output performs the NOR function of the inputs and is functionally complete for specifying a system; however, the author cleverly chooses to use another core in series with the complementing core. By allowing only one of the two cores to be set during an input phase, the load presented to a low-impedance, two-phase clock can be relatively symmetrical. Because the magnetizing current of the nonblocked core determines the series clock current, the input for a set or blocked Symmag element can be controlled from an inhibiting current source. The symmetrical loading and current inhibit technique distinguishes the Symmag element from other core logic circuits.

The price that is paid for the additional core complexity is also outweighed by its ability to produce an output which is the complement of the output of the inverted core; that is, the OR function as well as the NOR function is available at the output of a Symmag element.

The author's implication that the Symmag element does not have some of the basic limitations of speed, voltage and frequency variations of other core circuits is certainly stretching a point. Also operation at "several hundred kilocycles" cannot be termed "high-speed operation" even for core logic circuits.

Approximately half of the paper is devoted to the application of the Symmag element to the positioning control of a machine tool. Because of the sketchy nature of actual engineering data presented on the Symmag element, the part on machine control could have been better used in supplying the performance information necessary for a proper evaluation of the element.

Donal A. Meier
Electronics Div.
The National Cash Register Co.
Hawthorne, Calif.

R61-70 Magnetic Logical Elements for Automatic Control Circuits—N. P. Vasil'eva and N. L. Prokhorov. (English transl. of *Automation and Remote Control*, vol. 20, pp. 1601–1610, December, 1959.)

This paper is supposed to be a critical survey of "Basic Systems of Logical Elements Based on Magnetic Cores and Semiconducting Diodes." It is not. It is divided into two sections by the conclusions which appear in the middle of the paper. The first part of the paper is concerned with logic control circuits strongly related to computers: "series" logic circuits such as CYPAK, FERRACTOR, LOGIMAG, and "parallel" logic circuits such as those of DI/AN Controls, Inc. No mention is made of core-transistor, core-capacitor, or all-magnetic logic circuits. No logic circuits developed in the USSR are introduced. Even the coverage of core-diode circuits is incomplete, missing many of the interesting circuits which are seemingly ubiquitous in American journals. In fact, the survey seems to be based on a reading of a total of four articles of non-Russian origin. There is a striking similarity between Figs. 1–5, 22 and 23 of Vasil'eva and Prokhorov and Figs. 8, 7(b), 9(a), 9(b), 10(c), 11(a) and 10 of a 1957 article by Mathias,[1] which is one of the references the authors cite. (There is probably no significance in the fact that diodes which Mathias showed as "going to the right" have uniformly been changed to "going to the left.")

Allowing for the unavoidable quirks of twice-translated material, there are so many omissions and misleading statements in the descriptions of circuit operation that one should be cautious in accepting any conclusions the authors draw of the relative merits of the various circuits. In their descriptions, little or no mention is made of the importance of signal phases, clock pulse sequences, relative winding polarities, limitations of logical interconnection, etc. Unilateral transfer of information is attributed to input-output turns ratios alone, ignoring the essential nonlinear forward diode characteristic and the current threshold of the magnetic cores. The stacking of diodes to obtain logical AND gates with many inputs is said to be "theoretically unlimited." This is no more true in core-diode logic circuits than in any other diode logic net (unless one uses faulty "theory"). In their conclusions, the authors compare the various circuits in terms of the numbers of components necessary to form logical AND gates. Circuits which by nature form NOR or NAND gates, of course suffer in this arbitrary sort of evaluation.

In short, there seems to be little of value in the first part of this article that could not be obtained from the wealth of magnetic logic surveys by Lincoln,[2] Rajchman,[3] Clark,[4] Meyerhoff,[5] and myself.[6]

The second part of the article is concerned with conventional

[1] R. A. Mathias, "Static switching devices," *Control Engrg.*, vol. 4, pp. 67–94; May, 1957.
[2] A. J. Lincoln, "Ferromagnetic Core Logical Circuitry and its Application to Digital Computers," BRL, Aberdeen, Md., Rept. No. 911; August, 1955.
[3] J. A. Rajchman, "Magnetics for computers—a survey of the state of the art," *RCA Rev.*, vol. 20, pp. 92–135; March, 1959. See also, *Special Tech. Conf. on Nonlinear Magnetics and Magnetic Amplifiers Proc.*, Los Angeles, Calif., August, 1958; pp. 179–221.
[4] E. G. Clark, "A survey of solid-state logic circuitry," *Digest of Tech. Papers*, 1958 Transistor and Solid-State Circuits Conf., Philadelphia, Pa., February 20–21; pp. 49–50.
[5] A. J. Meyerhoff, "Digital Magnetic Circuits, Theory, Electrical Design, and Logical Design," Burroughs Corp., Detroit, Mich., Tech. Rept. TR59-27 (three-volume series), Contract Nonr-2145(00), ASTIA Documents AD-216961, AD-215-789, AD-215790; April, 1959. (Also available under one cover in recent publication by John Wiley and Sons, Inc., New York, N. Y.; 1960.)
[6] J. L. Haynes, "Logic circuits using square loop magnetic devices: a survey," IRE Trans. on Electronic Computers, this issue, pp. 191–203.

magnetic amplifier techniques. If this sort of thing interests you, the information is much more readable in its original form in Mathias' article.

John L. Haynes
Precision Instrument Co.
San Carlos, Calif.

R61-71 Thermal Propagation of a Normal Region in a Thin Superconducting Film and its Application to a New Type of Bi-Stable Element—R. F. Broom and E. H. Rhoderick. (*Brit. J. Appl. Phys.*, vol. 11, pp. 292–296; July, 1960.)

If a gradually increasing current is passed through a superconducting film containing a narrow region or flaw, the film in that region will become resistive, due to the magnetic action of the current, before the rest of the film does so. When this happens, Joule heating occurs in the resistive region. The resulting temperature rise causes the region adjacent to the flaw to become resistive, which in turn increases the Joule heating. A runaway process then occurs, and the resistive region spreads over the whole film. It is this process which is called thermal propagation.[1] If the current through the film is now reduced to a value which may be much smaller than the value at which thermal propagation occurs, the amount of Joule heating will become so small that the resistive region starts to collapse and the film returns to the superconducting state. It is possible to initiate thermal propagation, at currents smaller than that at which thermal propagation starts from a flaw, by temporarily creating a nucleus by passing current through a wire laid across the film at right angles to the current. The magnetic field due to current in this wire creates a nucleus in the film which can initiate thermal propagation.[1]

In the present paper, Broom and Rhoderick derive analytically the thermal propagation speed of an interphase boundary in an infinite strip. They find order of magnitude agreement between theory and experiment. Their analytical result agrees with that of an unpublished calculation carried out by the reviewer some years ago.

The authors point out that after a normal region has been initiated and is growing under the heating influence of a current, its growth velocity can be reduced to zero by reducing the heating current by just the right amount. They suggest that this phenomenon could be applied to a bistable storage element consisting of a superconducting film, carrying a subcritical continuous current I. Application of an "ignition" pulse would initiate a normal region (or regions) which would spread to an extent determined by the amplitude and duration of the ignition pulse. Application of an "extinction" pulse—I of duration comparable to the cooling time-constant of the strip would allow the strip to become and remain superconducting even when I is reapplied.

The authors present some results of an experiment on a thin film on mica carrying a steady current of 11 ma which could be "ignited" every 500 μsec with a 10-nsec current pulse of +22 ma followed 1 μsec later by a 10-nsec extinction pulse of −22 ma.

They find that the length required for the extinction pulse is very repetition rate sensitive. While recognizing this limitation and the disadvantage of the power consumption in the ON state, they suggest that "this bistable element could conceivably find some application in computer design." They cite as its advantages "extreme simplicity of construction, high speed of operation, low drive current and nondestructive readout in the form of dc voltage." While deprecating the "prevailing fashion of concocting a name for any device embodying a new principle," they nevertheless succumb to the temptation and suggest the name "calotron."

Although the calotron in its present form would appear difficult to use because it is a two-terminal element, this can be cured by making it into a four-terminal element using a transverse wire for nucleation.[1] Further, although the device is presently slow and dissipates a relatively large amount of heat (10^{-4} w), it may become faster and cooler if it is made sufficiently small. It seems eminently possible therefore that the calotron or one of its descendants will one day become important.

V. L. Newhouse
GE Res. Lab.
Schenectady, N. Y.

[1] J. W. Bremer and V. L. Newhouse, "Thermal propagation effect in thin superconducting films," *Phys. Rev. Lett.*, vol. 1, pp. 282–284; October 15, 1958.

R61-72 Proc. Symp. on Microminiaturization of Electronic Assemblies—Eleanor F. Horsey, Ed. (Hayden Book Co., New York, N. Y., 278 pp. +8 index pp.; 1959. $11.00.)

The editor has compiled a collection of papers presented at a DOFL[1] symposium on microminiaturization in the fall of 1958. As such, they represent the state of the art, at that time, of a field which is so dynamic as to be almost impossible to be adequately represented in this way. Nonetheless, it does serve as a good starting point for someone just entering the field in that it introduces the reader to many of the approaches which have been attempted in the past.

The book is compiled in six sections dealing with techniques, semiconductors, components, circuits, missile systems and microelectronics in industry. Of these, the best by far is the section dealing with techniques. These papers present work in fine-line etched wiring, survey of equipment adaptable to microminiaturization, application of vacuum evaporation techniques and interconnection of microminiature electronic subassemblies. This last section should have been expanded to give even greater attention to the problems of interconnecting the subassemblies. While it is recognized that there is still much to be done before a good solution to this problem is found, the fact that the book does not emphasize the importance of it points out a weakness which cannot be overlooked.

The section on semiconductors deals primarily with the problems of stability of semiconductors and two-dimensional packaging. Both of these areas have undergone considerable change since the book was published, leaving the reader far behind the present state of the art.

Most of the section dealing with circuits could have been deleted without much loss to the reader. While it is recognized that the design of circuits plays an important part in microminiaturization, I fail to see the need for a short course in circuit design in a book such as this. Only one paper really deals with the problems circuit designers encounter in microminiaturization.

The remaining two sections deal with the miniaturization of components such as magnetrons, antennas, indicator lamps, etc.

While the book does serve as a survey of the art in microminiaturization, it has not gone deep enough into any of the problems to make a newcomer aware of the severity of them or the extent of the work which remains to be done before anyone can claim success.

R. J. Domenico
IBM Corp.
Poughkeepsie, N. Y.

[1] Diamond Ordnance Fuse Labs. Symp., Washington, D. C. November 30–October 1, 1958.

F. Memories and Access Circuits

R61-73 Associative Self-Sorting Memory—R. R. Seeber. (*Proc. Eastern Joint Computer Conf.*, pp. 179–187, New York, N. Y.; December, 1960.)

This paper describes briefly the concept of associative storage and then demonstrates a technique for augmenting such a store so it becomes self-sorting. The scheme is built upon an associative store implemented in thin-film cryogenic circuits.

The technique employed uses simultaneous high-low-equal comparison circuits for all bits to determine the slot into which the word to be sorted is to be placed. Then the new word is inserted into a dummy register between the two appropriate words and the contents of the words above this register are shifted up into corresponding adjacent dummy registers. Finally, the contents of all dummy registers are shifted up one level to the appropriate working registers. All words, including the newly sorted word, are now in their appropriate positions.

The author accompanies his text with several diagrams of the system and cryogenic circuits to realize the system. He notes that the memory, as presented, allows for backward as well as forward sorting and he gives a time-step example of the latter.

The reviewer questions the practicality of adding the necessary hardware to an already powerful machine for the purpose of implementing self-sorting. The advantages gained for a limited subset (*e.g.*, single-word records only) of sorting operations do not seem to outweigh the disadvantages of having half of the words in the store sitting idle except for such operations.

Robert F. Rosin
Computing Ctr.
University of Michigan
Ann Arbor, Mich.

R61-74 A New Semiconductor Memory Element with Non-Destructive Read-Out and Electrostatic Storage—V. H. Grinich and D. Hilbiber. (1960 IRE WESCON Convention Record, pt. 3, pp. 34–41.)

The *p-n-p-n* diode has been used during the past several years as a two-state circuit element—conducting or nonconducting. One apparent shortcoming is the sensitivity of the breakdown voltage threshold to the rate of rise of the applied voltage. This paper describes a concept that takes advantage of this phenomenon, in a controlled form, to make use of the device as a memory element. In this case, both states are nonconducting.

If the junctions of a *p-n-p-n* diode are uncharged, a voltage "pulse" applied to its terminals with a certain rate of rise and amplitude will cause the diode to break down into its high-conduction state. However, if a well-chosen charge is first stored in the device, the same voltage "pulse" will not cause this breakdown. Further, when breakdown occurs, the charge is destroyed and when no breakdown occurs, the charge is regenerated. Thus, for a *p-n-p-n* diode with specified characteristics used as a memory element, three kinds of voltage pulses may be applied: 1) a WRITE-1 that is certain to cause breakdown because it has a fast rise and a large amplitude; 2) a WRITE-0 that is certain not to cause breakdown and to charge the device because of a slow rise and an amplitude less than the breakdown voltage; and 3) a READ having a rise time and amplitude that will cause breakdown if and only if the last applied pulse caused breakdown (no charge). The READ process is therefore nondestructive. This form of memory requires only enough power to regenerate charges lost by stray leakages.

The authors base their description of the concept on gross simplifications of the mathematics and they substantiate the operation with some experimental results. Reference is made to a mysteriously coded *p-n-p-n* diode: "typical commercially available four-layer diode." However, except for mention of a 100-kc *maximum* repetition rate on these units, the numerical results are based on composite units using transistors, and no mention is made of the *minimum* repetition rate necessary for charge regeneration. These results indicate that the rise time could be varied over a four-to-one range for satisfactory operation, but the amplitudes could vary over only a five-to-three range. In my mind, this precludes the use of coincident voltage techniques for addressing a memory cell in an array, relegating the composite device to small memories only. I have no information upon which to comment on what performance can be expected from "typical commercially available four-layer diodes."

Irwin Dorros
Bell Telephone Labs., Inc.
Murray Hill, N. J.

R61-75 A Vacuum Evaporated Random Access Memory—K. D. Broadbent. (Proc. IRE, vol. 48, pp. 1728–1731; October, 1960.)

Magnetostatic energy resulting from the divergence of magnetization has been employed in the operation of multipath ferrite devices; this paper reports the results of a multipath magnetic film memory element employing the same magnetostatic energy principle. The resulting memory element has switching times as low as 0.03 μsec, wide latitudes in selection currents, and the small volume inherent in film devices.

Although progress in material uniformity probably makes this relatively complicated memory element structure unnecessary for most applications, the fabrication techniques described may be very significant, particularly in magnetic film logical devices as well as memories.

High-performance film logical devices generally require thin conductors and thin dielectrics. The paper reports vacuum deposition techniques by which apparently the required thin dielectrics and conductors can be made with considerable ease. Consequently, these techniques may prove important in the development of magnetic film logical networks.

Arthur V. Pohm
Elec. Engrg. Dept.
Iowa State University
Ames, Iowa

R61-76 Magnetic Film Memories, A Survey—A. V. Pohm and E. N. Mitchell. (IRE Trans. on Electronic Computers, vol. EC-9, pp. 308–314; September, 1960.)

This paper serves the useful purpose of acquainting those unfamiliar with the art with a variety of possible magnetic film memory configurations and relating them to fundamental magnetic properties. It suffers from the limitations imposed by compressing a considerable body of physics and engineering work into a brief survey. The first section provides a summary of magnetic properties and a description of the uniaxial anisotropy model. Unfortunately, the authors complicate the discussion by introducing energy contributions due to the tilt of M out of the plane, which are of no significance, but do not show how the ideal B-H loops and threshold curve may be derived from the simple model.

The section on operating modes is somewhat vague because of the qualitative nature of the statements about speed and material requirements for different systems. In particular, the statements which imply that large three-coordinate memories can be built at some sacrifice in operating speed with "good magnetic material uniformity" are liable to serious misinterpretation. To the reviewer's knowledge this has never been satisfactorily demonstrated. In discussing memory modes using wall switching, the authors draw the unjustified conclusion that a system which relies on two parameters (H_c and H_k) necessarily has tighter tolerances than one which makes use of only one (H_c). This is not necessarily true, and for this example, in particular, is extremely doubtful since the rotational memory using perpendicular word fields and parallel digit fields has wider tolerances than a memory using longitudinal fields alone. For word-organized, nondestructive read-out memories employing perpendicular fields the authors state that "the transverse field must be smaller than the creep threshold, and the longitudinal field must be less than the coercive force, while the combination of both should produce simple rotational switching." In a practical system, one would not use the same field value for reading and writing, but instead, a large field for writing and a field below creep for reading. Thus, the material requirements for writing are identical to those of the DRO system and the only remaining disadvantage would be the decrease in signal imposed by the creep threshold.

In the section on material problems, relatively unimportant factors such as control of thickness are discussed, but the crucial problems of skew and dispersion of the easy axis direction are unmentioned.

J. I. Raffel
M.I.T. Lincoln Lab.
Lexington, Mass.

R61-77

1) A Class of Optimal Noiseless Load-Sharing Matrix Switches—R. T. Chien. (*IBM J. Res. & Dev.*, vol. 4, pp. 414–417; October, 1960.)

2) New Developments in Load-Sharing Switches—G. Constantine Jr. (*IBM J. Res. & Dev.*, vol. 4, pp. 418–422; October, 1960.)

Improved techniques for reducing the number of inputs required to drive a load-sharing matrix switch are presented in these papers. Chien makes use of winding matrices in which the winding pattern is obtained by the construction of a special class of orthogonal matrices. This method requires a minimum number of input drivers for a noiseless load-sharing matrix switch. Constantine, on the other hand, departs from noiseless load-sharing matrix switches and employs binary code sequences to construct his winding patterns. This method permits small excitations on some of the unselected cores in order to reduce the number of input drivers required.

A major section of Chien's paper consists of a summary of Paley's work on the construction of orthogonal matrices. This section is tedious to follow and will be appealing to only a limited number of readers. Consequently, since the concept presented can be understood without it, it is suggested that most readers skim it. Also, it should be pointed out that while the scheme requires a minimum number of input drivers, the logic required to select the input drivers will be relatively complex. Consequently, the net improvement of this scheme over previous noiseless core load-sharing schemes will depend upon its application.

Constantine's paper presents a scheme which is more easily implemented in practice. Although excitation is permitted on unselected cores, its magnitude can be adjusted to whatever value is required depending upon the amount of decoding employed. Thus, by proper choice of the core material and the amount of decoding acceptable, low noise outputs can be obtained. This paper is a good extension of core load-sharing techniques and is presented in a clear yet concise manner.

Charles J. Vincelette
Bell Telephone Labs., Inc.
Murray Hill, N. J.

R61-78 Distributed Parameter Aspects of Core Memory Wiring—J. S. Eggenberger. (*Proc. 5th Annual Conf. on Nonlinear Magnetics and Magnetic Amplifiers*, Philadelphia, Pa., October 26–28, 1960; pp. 6–14.)

This paper attempts to determine the transient behavior of one line of a conventional core plane by operational analysis of an idealized model consisting of an open-wire transmission line furnished with lumped impedances representing the cores and crossed-wire capacitances. To particularize the model, the author offers an experimentally determined parallel *RL* equivalent circuit for a ferrite core under half-select conditions. An empirical formula based on measured data for the capacitance between a pair of crossed cylindrical conductors is also presented in an appendix. Unfortunately, this substantially summarizes the useful contents of the paper, since the lengthy transient analysis performed on the resulting model is conceptually shaky and mathematically inadequate, leading to physically absurd and incorrect answers.

As an exercise in linear-transmission-line analysis, the required transient behavior of the proposed model may be obtained by straightforward, albeit somewhat tedious, application of conventional Laplace Transform techniques. Instead, the author chooses a devious procedure in which he first obtains unit step approximations in the p domain for $Zo(p)$ and $\gamma(p)$ which are free from square roots. The resulting expressions can at best be regarded as of doubtful validity, since no error analysis is attempted. (The new expressions correspond in fact to a line which cannot be realized with two-terminal impedance elements.) Using the approximate expression for $\gamma(p)$, the author then attempts to obtain the voltage response of the line to a unit step of voltage at the input. To do this, he expands $\exp[-\gamma(p)x]$ in a double-power series which was "programmed for the IBM Data Processing System for a range of values of X and T. Terms smaller than 0.01 were ignored."

It is a pity that the author went to all this trouble, since simple closed-form expressions for the desired function can be obtained in various ways. For example, I find that:

$$E(X, T) = e^{-X}\left[1 + 2\int_0^{\sqrt{XT}} e^{-u^2/X} I_1(2u)du\right]$$

is one of several equivalent forms for the voltage along the line. X and T are the normalized variables employed by the author and the delay term has been suppressed.

Something went wrong with the author's machine computation. Apparently the terms smaller than 0.01 contained most of the essential features of the solution (at least for large X) since the computed data presented in Fig. 3 are obviously incorrect. The figure shows waveforms of voltage vs time for various distances down the line in which the wave front becomes steeper with increasing distance. This is remarked upon in the text, but it is physically impossible for the linear line treated. Similarly, the wiggles which begin to appear in the voltage waveform at larger distances ($X=10$ in Fig. 3) cannot exist, since the above equation clearly shows that $E(X, T)$ is a monotonically increasing function of T for fixed X.

Subsequent mathematical approximations and graphical data presented in the paper depend upon the incorrect machine solution and consequently cannot be taken seriously. Incidentally, the further approximations for current and voltage which the author tried to obtain may be rather easily deduced from the closed-form expression offered above.

Milton W. Green
Computer Techniques Lab.
Stanford Res. Inst.
Menlo Park, Calif.

G. Programming and Numerical Methods

R61-79 Programming Computers to Play Games—Arthur L. Samuel. ("Advances in Computers," Franz Alt, Ed., Academic Press, New York, N. Y., vol. 1, pp. 165–192; 1960.)

This is an excellent review paper. It traces the development and gives the current status of using computers to play games. The computer actually plays one side of the game as opposed to being used only to provide an environment, as for "business" or "war games." Games are convenient for studying the simulation of human intellectual processes with a computer since they "retain many of the essential characteristics of real-life problems and eliminate many of the worrisome complications."

Without using technical jargon, the author gives a very good idea of the problems one meets, as well as the present accomplishments in writing programs for a computer to learn to play a game. The paper reports some very real advances (most if not all of which are the result of hard work by people) in the field of "artificial intelligence." This stands out in contrast to all the nonsense on this subject which finds its way into print.

D. W. Hagelbarger
Bell Telephone Labs., Inc.
Murray Hill, N. J.

R61-80 Applications of Graphs and Boolean Matrices to Computer Programming—Rosalind B. Marimont. (*SIAM Rev.*, vol. 2, pp. 259–268; October, 1960.)

Many aspects of instruction sequencing in digital computer programs can be represented by directed graphs (digraphs) in which the vertices correspond to instructions, and the directed edges specify the possible sequences of execution. This paper gives an elementary survey of the appealing and rapidly developing theory of digraphs, and attempts to point out some applications to programming.

The fundamental tool of analysis is a square zero-one matrix A, corresponding to a given graph, in which $a_{ij}=1$ if the graph has a directed edge from vertex i to vertex j. The sum, product, and element-wise product of such matrices are defined, and the powers of a matrix are related to directed paths in the associated graph; in particular, it is shown that the matrix A is nilpotent if and only if the associated graph has no cycles. A *reachability matrix*

$$R = \sum_{i=1}^{\infty} A^i$$

is defined, and its application to the detection of certain "blind alley" programming errors is indicated. Several other matrix manipulations of less pertinence to programming are also mentioned.

Up to this point, the paper is a readable digest of elementary computations with Boolean matrices. The ensuing brief discussion of applications to programming is less successful. The author distinguishes between an operation which may begin when any of its predecessors have been completed and one which must await the completion of all its predecessors, and remarks that this distinction is not made in usual flow chart notation. There is also a discussion of the dilemma that "a closed loop in a set of precedence relations is an inconsistency, and yet correct flow charts consist mainly of loops. . . ." Such problems are superficial, and easily resolved. Far more serious problems, not mentioned in this paper, obstruct the application of graph theory to programming. For example, a program may modify its own instructions during execution, possibly necessitating a change in its graph-theoretic description. Moreover, even when a graph ade-

quately describes the possible sequences of execution of a program, many applications require knowledge of the conditions determining the particular sequence to be followed. Such genuine difficulties must be given serious consideration in any application of graph-theoretic models to programming.

Richard M. Karp
IBM Res. Ctr.
Yorktown Heights, N. Y.

R61-81 Digital Computers and Nuclear Reactor Calculations—Ward C. Sangren. (John Wiley and Sons, Inc., New York, N. Y.; 1960. 208 pp.+xi pp. $8.50.)

This book is a survey and instruction manual for neophytes concerning the use of digital computers in nuclear reactor design problems. The first half of the book yields a short history of computers and their application to reactor design, an elementary description of general purpose digital computers, a programming primer, and a heuristic discussion of some elementary methods of numerical analysis. The second half consists of examples of reactor design mathematical problems and their reduction to computational procedures. The first and simplest of these problems, which is considered in Chapter 5, "A Code for Fission-Product Poisoning," is displayed in complete detail from physical assumptions and mathematical formulation through coding and debugging to samples of a standard punched card form for input data and a facsimile of the code's printed output. Since the bulk of reactor design calculations deals with the physics of reactors, it is natural that the author's examples are largely selected from the problems of reactor physics. A short bibliography cites works ranging from treatises, for those who wish to examine a topic in greater depth, to reports on codes in daily use, for those who may discover that their design problem has already been reduced to a computer code. An adequate index is provided.

Physically, the book is well made and easily read. The text seems reasonably free of typographical errors. One exception to this occurs on pp. 184–187, where parentheses appear and disappear with annoying inconsistency in the kernel of the integral defining the shape functions for Doppler-broadened resonance curves.

Since the contents of this book are generally elementary, accurate and clearly presented, two questions remain to be answered. What is an appropriate audience for the book? Where does the book stand in its genre?

Let us consider the first question. With a few deletions or changes from appeal to reactor problems for motivation, the first half of the book could stand alone as an introduction to digital computers for the uninitiated engineer, whatever his field. The second part of the book is of immediate utility to those who are familiar with the problems of reactor design. They will find a clear description of computational attacks on some problems and be able to generalize from these examples. The scientist who is not involved in reactor design may see that the examples, which are discussed in the nomenclature of reactor theory, are common to many branches of engineering. Computing experts will find the book a useful text for instructing reactor engineers in the techniques of numerical calculation.

Second, let us consider the position of this book with respect to others of its ilk. For years there has existed a submathematical library of handbooks and manuals which purport to explain without rigor certain mathematical processes and applications. Recently, digital computers and numerical analysis have been added to the list of topics covered by these much used texts. Sangren's book is a respectable member of this class. This reviewer agrees with the author's choice of recognizing the existence of certain numerical problems without naming or defining them rather than offering an inadequate definition.

This book is needed in the library of every nuclear reactor design group. It will propagandize the possibilities of digital computers for nuclear reactor calculations, render communication easier between the engineers and computing personnel of such groups, or induce the engineer to become competent to deal with computational problems. The first of these alternatives is probably most important but waning. There still are many scientists unaware of the ease of using a truly powerful mathematical instrument, the digital computer, to check, evaluate, and implement their mathematical models of physical situations. It remains to be seen whether a reader who is not interested in reactor design will have the patience or perspicacity to seriously consider programming examples which, though quite general, are couched in the terminology of reactor designers.

R. van Norton
A.E.C. Computing and Appl. Math. Ctr.
Inst. of Math. Sciences
New York University
New York, N. Y.

R61-82 Finite-Difference Methods for Partial Differential Equations—G. E. Forsythe and W. R. Wasow. (John Wiley and Sons, Inc., New York, N. Y.; 1960. 444 pp.+x pp.)

This book is not for the beginner. Primarily, the authors treat those aspects of partial differential equations and their finite-difference representations which deal with stability and truncation errors. In some places, the exposition actually takes the form of definition, theorem, and proof. Even where the exposition is less formal, the authors display a precision in their writing which combines with their interest in detail to produce a work which engineers unfamiliar with the subject will find difficult to read. (One is tempted to speculate that the present formats enforced by mathematical journals may render mathematicians incapable of communicating with other human beings after comparatively few years.) This is the more unfortunate because the authors have considerable to say on the subject of partial differential equations which is both useful and interesting.

The first 84 pages discuss hyperbolic equations in two independent variables—the bulk of the exposition being devoted to integration via characteristics. This is followed with 50 pages on parabolic equations and then 236 pages on elliptic equations—reflecting the relatively greater amount of knowledge which has been accumulated over the years in potential problems. The book concludes with 34 pages on propagation problems in more than two independent variables.

Much of the material appearing in this book has only been available in the literature of the last 10 years. Theories of successive overrelaxation, alternating direction methods of solving diffusion and potential problems, suppression of high-frequency waves in meteorological forecasting, and determination of bounds for eigenvalues in membranes are examples of topics which are here collected for the first time. This reviewer can recommend the book as a source of considerable valuable information, although it may take quite a bit of effort to sort the information into its proper perspective. The extensive bibliography should prove useful.

A more accurate title for the book might be "The Numerical Analysis of Partial Differential Equations and Their Finite Difference Representations." We still need a book which points out what the major difficulties are and how the various methods (examined in detail here) fit into the general framework.

Forman S. Acton
Princeton University
Princeton, N. J.

H. Analog Systems

R61-83 Analog Computation in Engineering Design—A. E. Rodgers and T. W. Connolly. (McGraw-Hill Ser. in Information Processing and Computers, McGraw-Hill Book Co., Inc., New York, N. Y.; 1960. 443 pp.+6 index pp.+x pp. $16.00.)

This book is the first in the analog computer field that can be described as being advanced. Its emphasis is on analog computer applications rather than on computer design or operation. The material on applications is contained in the last two-thirds of the book. In an attempt to make the book self-contained, the early chapters contain a brief introduction to analog computers and a review of engineering mathematics.

Chapter 1 contains a brief history of analog computing and a discussion of the "analog" approach to problems. Discussed in Chapter 2 are all the basic computing components, including input-output equipment, and computer control functions. In the discussion of

high-gain amplifiers, no mention is made of "implicit" techniques. A discussion of "implicit" techniques would lead naturally to the division circuit and also to the computer circuits for solving algebraic equations in Chapter 6.

The mathematical background of linear analysis and the simulation of differential equations is taken up in Chapter 3 of over 80 pages. The discussion of Bessel's Equation is marred by several minor errors. In Fig. 3-60 the term

$$\left(\frac{100}{5\tau}\right)\left(\frac{10\dot{y}}{20}\right)$$

should have a minus sign, and in Fig. 3-61 the output from multiplier 2B requires a gain of 10. In the discussion of the improved circuit for solving Bessel's Equation, it is not made clear that the improved behavior is at the expense of an approximation lasting for 0.1 second. On page 137, the statement is made that the circuit accounts properly for the values of the first three derivatives of y. What is meant is that at $\tau=0$, the value of y and its first two derivatives are properly accounted for.

Covered in the next few chapters are general techniques, solution of partial differential equations, algebraic and other mathematical problems, and nonlinearities. Also included are chapters on random signals and a very good chapter on the adjoint computing technique. For the first time, I really understood the adjoint technique and its power in the study of linear time-varying problems. The chapter on random noise is also very good, although lacking a discussion of the time required for making low-frequency statistical measurements.

An excellent procedure for the setting up and solving of problems is outlined in Chapter 10. It cannot be overemphasized that the majority of errors in obtaining solutions by analog means are human ones and not due to malfunctions of the computer. A definite procedure, perhaps however not as detailed as the one presented, is a necessity for eliminating possible errors and simplifying the location of other errors. I take minor exception to recommendation that . . . "time- scaling be reduced to a purely mechanical operation on the computer circuit diagram." For example, in a radar system simulation where it is necessary to slow time down by a factor of 10^6, the numbers are easier to work with if the equations are first converted to the new time scale.

The last six chapters cover specific analog computer applications, in a variety of fields, from nuclear-reactor engineering to guided missile problems. Particularly good are the chapters on "Nuclear Reactor Engineering" and "Problems From the Aircraft Industry." In general, all the application chapters follow the same format: a discussion of the physical problem with a derivation of the basic equations and the computer diagrams for the solution of the equations. Missing in practically all the cases are any actual computer results and a discussion of computer errors and solution accuracy.

In the chapter on "Electrical Design Problems," the problem of simulating ideal, linear, passive, electric networks is discussed. This is a frustrating problem, as quite frequently, application of straightforward techniques leads to a simulation containing redundant integrators, *i.e.*, more integrators than energy storage elements in the original network. With the redundant integrators, the solution will probably not bear much resemblance to the true solution. The example given is first solved using standard loop currents and results in a redundant integrator. Nodal analysis is then used and the redundant integrator is eliminated. It is implied that the use of nodal analysis eliminates the danger of redundant integrators and this is not true in general. Otterman[1,2] has given rules for writing the network equations in order to avoid redundant integrators.

An extensive bibliography is included, covering the period from 1950 to the middle of 1958, in chronological order.

Scattered through the book are a few rather vague or mysterious statements. On page 11 in discussing the negative gain of an operational amplifier, we find . . . . "This change in gain is necessary, for the impedance Z_f which surrounds completely the amplifier from output to input grid has to produce negative feedback." From a discussion of transient response on page 390 . . . "Transient response (impulse response) may be obtained by exciting the system with an initial condition. Similar in usefulness to transient response for many purposes, although not identical mathematically, is the response to a step input (Fig. 16-7). Such an input will produce an output which is the sum of the impulse response and steady state response to the step."

In summary, the book has a few flaws, but it can still be recommended to any analog computer engineer who is interested in the broader aspects of the analog computer field.

Arthur D. Bridgman
Electronic Defense Labs.
Sylvania Electric Products Inc.
Mountain View, Calif.

[1] J. Otterman, "How to avoid extra integrators when simulating RLC networks," *Control Engrg.*, vol. 4, pp. 111–114; November, 1957.
[2] J. Otterman, "On the loop- and node-analysis approaches to the simulation of electronic networks," IRE Trans. on Electronic Computers, vol. EC-7, pp. 199–206; September, 1958.

Abstracts of Current Computer Literature

(THROUGH JANUARY, 1961)

These abstracts and the associated subject and author indexes were prepared on a commercial basis by Cambridge Communications Corp., Cambridge, Mass., under the direction of Dr. Geoffrey Knight, Jr., who also publishes the abstract journal "Solid State Abstracts," and the card services "Solid State Abstracts on Cards" and "Computer Abstracts on Cards."

—The Editor

CONTENTS

0. GENERAL

1309
European Electronic Data Processing—A Report on the Industry and the State-of-the-Art by I. L. Auerbach (Auerbach Electronics Corp.); PROC. IRE, vol. 49, pp. 330–348; January, 1961.

Information processing activities in Western Europe are reviewed. Newly-created companies and many of the established business machine and electronic firms are entering the field. A first-hand tour of major industrial and academic groups in Western Europe reveals important progress in equipment design and manufacturing techniques, as well as significant advanced development work. The United States is ahead of Western Europe, due primarily to a greater over-all research and development effort. However, European laboratories are developing new techniques and products for the world market, ranging from peripheral equipment to complete information processing systems. The survey of European activities is presented in three parts: an introduction, details of some important technological developments under way, and a detailed review of the characteristics of European computing systems. The technological developments include fixed high-speed memories, magnetic thin films, random-access memories, pattern recognition, learning machines, hydraulic logic, and problem-oriented languages.

1310
Notes on the State of Digital Computing in the USSR by L. Clarke (Elliot Bros. Ltd.); *Computer J.*, vol. 3, pp. 164–167; October, 1960.

A visit to the computer installations in Russia in connection with the exhibition of a National-Elliot 803 Computer at the Scientific Instrument Manufacturers Association's exhibition in Moscow is described. General descriptions of the BESM2 and URAL 1 computers are provided. A major Russian interest is said to be the digital transduction of physical measurements such as temperature, pressure, etc.

1311
A Logical Machine for Measuring Problem Solving Ability by C. R. Langmuir (Psychological Corp.); *Proc. Eastern Joint Computer Conf.*, pp. 1–9; December 13–15, 1960.

The magnitude of costs incurred by assigning unsuccessful or even marginal personnel to tasks involving EDP systems design and programming justifies a much greater effort in the selection of personnel than the use of conventional aptitude tests implies. A small desk-top machine named the Logical Analysis Device is described, its logical organization is explained, and its operation as a method of observing and testing an individual's problem-solving abilities is illustrated with slides. Some comment describing the wide variation of performance among several hundred college graduates employed in various professions is included, but the principal emphasis is given to data pertaining to the performance characteristics of persons in computer and data-processing activities. The application of the device is clearly indicated at the point of evaluating final candidates for assignment to tasks requiring a high order of logical and analytical talent.

1312
Impact on Industry of Engineering Graduates Who have Used Computers in the Classroom by D. L. Katz and E. I. Organick (University of Michigan); *Proc. NEC*, vol. 16, pp. 316–322; 1960.

Shortly, engineering graduates will come to industry with a demonstrated ability to program problems on the digital computer and to employ analog computers to solve engineering problems. Some implications of this innovation in engineering education are pointed out for engineering management.

1313
The Computer Revolution in Engineering Education by R. E. Machol (Purdue University); *Proc. NEC*, vol. 16, pp. 305–307; 1960.

Experiences at Purdue University with a small digital computer on which thousands of undergraduates have been taught to program within the past year are described. A new approach to the teaching and administration of computers is required, as is a new approach to almost every aspect of the engineering curriculum. The shape of things to come is briefly examined.

1314
The Organization of a University Computing Centre by R. A. Buckingham (University of London); *Computer J.*, vol. 3, pp. 131–135; October, 1960.

Three separate areas of responsibility for a university computing center, namely the provision of a computing service for the university, the education and training of programmers and users, and research for both computer design and application, are designated. Since demands made on university computing centers are increasing, allocation of computer time must be carefully organized. Nonetheless, research must not be stultified by not allowing untried methods of problem solution, which may result in some wastage of computer time. "Open" shop operations are a necessity, so that researchers may program their own problems.

1315
Cope (Console Operator Efficiency Program) by D. Farbman and R. Kefover (Computer Usage Co. Inc.); *Commun. Assoc. Comp. Mach.*, vol. 3, pp. 661–662; December, 1960.

A program designed both to test the skill of a console operator and to act as a training aid for new operators is described. The program is divided into three phases of increasing complexity, and nearly every situation likely to require the operator attention is simulated on the program. The program described was written for an IBM 705 with two tape-record coordinators, but similar programs could be developed for other machines.

1. LOGIC AND SWITCHING THEORY

1316
Computational Aids for Determining the Minimal Form of a Truth Function by R. Prather (San Jose State College); *J. Assoc. Comp. Mach.*, vol. 7, pp. 299–310; October, 1960.

An algorithm for determining the minimal form of a truth function expressed in canonical form is described. All calculations are performed on decimal numbers obtained from binary-decimal conversion of the terms of the Boolean function. Computational aids to adapt the algorithm to the solution of large-scale problems on a computer are presented.

1317
Minimization over Boolean Trees by J. P. Roth (IBM Corp.); *IBM J. Res. & Dev.*, vol. 4, pp. 543–558; November, 1960.

An algorithm for the general problem of logical design of circuits with one output and no feedback is presented. The problem may be characterized as that of finding the minimum-cost Boolean tree constructed from a given set of building blocks which realizes a given Boolean function. A special case is the classical logical problem of finding a functional expression for a given logical function which uses a minimum number of conjunctions, disjunctions, and negations. A version programmed on an IBM 704 Computer is claimed to be efficient for 8 or fewer variables.

1318
The Simplification of Multiple-Output Switching Networks Composed of Unilateral Devices by G. C. Vandling (General Electric Co.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-9, pp. 477–486; December, 1960.

It is shown that two-level (no more than two gates in cascade) multiple-output switching networks composed of unilateral switching devices such as diodes can be simplified or minimized in much the same manner as single-output networks. This is accomplished by extending the notation and techniques used in the simplification of two-level single-output switching networks to multiple-output switching networks. A simple procedure for identifying multiple-output prime implicants is devised and, as a final result, an algorithm is presented which can be used to minimize the switching network corresponding to a number (q) of given Boolean expressions of the same variables. This algorithm is based on the Quine rules but has been modified to take advantage of the so-called "don't care" conditions which occur because some inputs are forbidden or because some outputs are of no concern. This algorithm can readily be programmed on a digital computer if desired.

1319
Arbitrary Boolean Functions of N Variables Realizable in Terms of Threshold Devices by O. B. Stram (Burroughs Corp.); PROC. IRE, vol. 49, pp. 210–220; January, 1961.

A method for the logical design of single-stage, combinatorial switching circuits of n variables is presented. This method is applicable to circuits composed of threshold devices, such as magnetic cores, transistors with Kirchhoff adder inputs, parametrons, etc. A study of the constraints imposed by the form of the input portions of the threshold devices leads to the definition of certain classes of functions which are physically realizable in a single device. By the use of this method, arbitrary switching functions of as

many as seven variables have been easily designed by hand computations.

1320
Minimal Complete Relay Decoding Networks by E. F. Moore (Bell Telephone Labs.); *IBM J. Res. & Dev.*, vol. 4, pp. 525–531; November, 1960.

It has always been assumed that the $2(2^n-1)$ contacts used in the standard n-relay tree network is the smallest number of contacts with which such a network can be made. A proof of this result, providing that no sneak paths are allowed, is given. This is in contrast to a result due to Lupanov, who has shown that for $n \geq 5$ it is possible to save contacts by permitting sneak paths. Further theorems about circuits analogous to the complete relay decoding network are proved.

1321
Characterizing Experiments for Finite-Memory Binary Automata by A. Gill (University of California); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-9, pp. 469–471; December, 1960.

The characteristics of a discrete automaton with a finite memory can be determined by an experiment of a finite length. The properties of such experiments are discussed, and methods for their optimal construction are presented. Specific results are given for binary-input automata with the memory ranges 0, 1, 2, 3 and 4.

1322
Connective Properties Preserved in Minimal State Machines by S. Ginsburg (System Dev. Corp.); *J. Assoc. Comp. Mach.*, vol. 7, pp. 311–325; October, 1960.

The question of what connective properties of machines are preserved in equivalent minimal-state machines is discussed. It is shown that the property of being strongly connected is preserved in at least one minimal-state machine. The preservation of stable states and the connective properties of minimal machines isomorphic to each other are also considered and various theoretical results are derived.

2. DIGITAL COMPUTERS AND SYSTEMS

1323
Flow Table Logic by P. R. Low and G. A. Maley (IBM Corp.); PROC. IRE, vol. 49, pp. 221–228; January, 1961.

The interconnecting of elements in proposed microminiature systems using packing densities on the order of 10^6 active elements per cubic foot is considered. A new theory of machine organization whereby rather large sequential circuits are implemented directly from their flow-table representations is suggested to solve this problem. This method does require more active elements, but the interconnection problem is greatly reduced.

1324
Parallel Computing with Vertical Data by W. Shooman (System Dev. Corp.); *Proc. Eastern Joint Computer Conf.*, pp. 111–115; December 13–15, 1960.

A novel technique called vertical data processing (VDP) for the manipulation of data on digital computers is presented. Multiple data are processed simultaneously one bit at a time using Boolean operations. Several classes of problems appear adaptable to this technique. A hypothetical VDP computer which embodies both VDP as well as conventional techniques is proposed, and its advantages are discussed.

1325
Impact of Automation on Digital Computer Design by W. A. Hannig and T. L. Mayes (General Electric Co.); *Proc. Eastern Joint Computer Conf.*, pp. 211–232; December 13–15, 1960.

The impact of design automation techniques upon the design, construction, and maintenance of digital computers is discussed. Specific items described include: 1) the logician's use of these programs as a design tool; 2) the use of the documents produced by these programs; 3) the effect that the use of these programs and program-produced documents has upon the human organization using them. The use to which automation programs were put in the design of presently operating digital computers, starting with Boolean equation input data and ending with factory release information, is also considered.

1326
Use of a Digital/Analog Arithmetic Uni Within a Digital Computer by D. Wortzman (IBM Corp.); *Proc. Eastern Joint Computer Conf.*, pp. 269–282; December 13–15, 1960.

The use of a digital/analog arithmetic unit in order to increase the computational power of digital computers is discussed. In some problems the inherent high accuracy of digital computers is unwarranted either because the input digital information is limited in accuracy or because the input information is in analog form. It is in these instances that high speed, ease of programming, and ability to operate on combined analog and digital information of the digital/analog arithmetic unit may be welcomed.

1327
High-Speed Arithmetic in Binary Computers by O. L. MacSorley (IBM Corp.); PROC. IRE, vol. 49, pp. 67–91; January, 1961.

Methods of obtaining high speed in addition, multiplication, and division in parallel binary computers are described and then compared with each other as to efficiency of operation and cost. The transit time of a logical unit is used as a time base in comparing the operating speeds of different methods, and the number of individual logical units required is used in the comparison of costs. The methods described are logical and mathematical, and may be used with various types of circuits. The viewpoint is primarily that of the systems designer, and examples are included wherever doing so clarifies the application of any of these methods to a computer. Specific circuit types are assumed in the examples.

1328
Binary Arithmetic by G. W. Reitwiesner (Aberdeen Proving Ground); *Advances in Computers*, Academic Press, Inc., New York, N. Y., vol. 1, pp. 231–308; 1960.

Procedures for the performance of binary arithmetic in the fewest number of sequentially performed steps or the fewest number of steps requiring specialized equipment are developed. Algorithms for the five basic arithmetic operations of addition, subtraction, shifting, multiplication, and division are provided. The square-root operation is briefly considered. Problems encountered in rounding, scaling, and multiple precision are also discussed. Floating-point operations and significance receive little consideration. The discussion is on an abstract plane.

1329
Multiple Precision Arithmetic by D. A. Pope and M. L. Stein (University of Minnesota); *Commun. Assoc. Comp. Mach.*, vol. 3, pp. 652–654; December, 1960.

A method of representing n-tuple precision numbers as n-digit numbers in a radix, each of whose digits occupies a whole order of precision, is discussed. The ability of the associated computer is assumed to be limited to forming the sum or difference of two such large radix digits, the two-digit product of a pair of such digits, and the single-digit quotient of a two-digit dividend and a single-digit divisor. Algorithms for addition and multiplication are straightforward. Division proceeds by recursively estimating quotient digits. Bounds on the required number of iterations are established.

1330
High-Speed Transistorized Adder for a Digital Computer by F. Salter (Argonne Natl. Lab.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-9, pp. 461–464; December, 1960.

An adder which has been developed for the floating indexed-point arithmetic unit, FLIP, to be used in conjunction with GEORGE, the existing computer built at Argonne National Laboratory, is described. The logic of the high-speed adder and the special circuits required are presented. The adder is parallel, and its high speed is made possible by reducing the carry propagation time. Each bit of the adder contributes one transistor to make up a tall AND gate which reduces the carry propagation time to 0.2 μsec. Using this high-speed carry propagation and rather common RCTL transistor circuitry, it is possible to complete an addition in less than 0.25 μsec.

1331
A Parallel Arithmetic Unit Using a Saturated-Transistor Fast-Carry Circuit by T. Kilburn, D. B. G. Edwards, and D. Aspinall (Manchester University); *Proc. IEE*, vol. 107B, pp. 573–584; November, 1960.

A transistor switch technique which is of particular importance in applications where a large number of switches have to be connected in series and where the propagation time of information through these switches has to be a minimum is described. Its use has been demonstrated in parallel addition with an addition time over 24 digits of 200 mμsec. It is shown how the technique, which is quite reasonable, can be used in conjunction with more conventional logical circuits to provide a simple arithmetic unit.

1332
Statistical Analysis of Certain Binary Division Algorithms by C. V. Freiman (IBM Res. Ctr.); PROC. IRE, vol. 49, pp. 91–103; January, 1961.

Nondeterministic extensions of the nonrestoring method of binary division have been described by MacSorley. One extension requires that the magnitudes of the divisor and partial remainders be "normal," *i.e.*, in the range (0.5, 1.0). This leads to a time improvement of more than two relative to conventional nonrestoring methods. Other extensions involve the use of several divisor multiples (or trial quotients). A Markov chain model is used to analyze these methods. Steady-state distributions are determined for the division remainder and performance figures based on both this steady-state distribution and a random distribution are calculated. These are compared with the results of a computer simulation of 2^{14} randomly-chosen division problems using two specific methods of division.

1333
Statistical Analysis of Logic Circuit Performance in Digital Systems by E. Nussbaum, E. A. Irland, and C. E. Young (Bell Telephone Labs.); PROC. IRE, vol. 49, pp. 236–244; January, 1961.

A digital system usually contains large numbers of relatively few basic circuit configurations. The over-all performance of such a system is largely limited by the characteristics of these building-block circuits. If these characteristics (speed, logical gain, noise margin) are treated on a statistical basis as opposed to a worst-case approach, substantial improvements in logical design flexibility may be attained. This paper reviews methods for combining statistical distributions of a set of parameters to obtain the distributions of the performance characteristics. Both algebraic and numerical (Monte Carlo) methods are considered. Transistor-resistor logic circuits are then analyzed in more detail. Resultant distributions of propagation delays and logical gain are obtained as functions of circuit parameter distributions, logical configuration and temperature. Comparisons with worst-case design figures are made to indicate the extent to which statistical techniques improve predicted performance. In addition, some new circuit configurations, proved usable by statistical analysis, are shown to lead to greater economy and reliability.

1334
The Organization and Program of the BMEWS Checkout Data Processor by A. E. Miller (Auerbach Electronics Corp.) and M. Goldman (RCA); *Proc. Eastern Joint Computer Conf.*, pp. 83–96; December 13–15, 1960.

The organization of the automatic checkout data processor (CDP) for the Ballistic Missile Early Warning System is discussed and the unusual structure of the CDP program is described. The CDP has separate memories; one for storing constants and instructions, and one for storing data. The means for jointly using these two different types of memories, while maintaining the flexibility associated with single-memory machines, is explained. The tailored features of the CDP, efficiently handling its unique problems, are emphasized. They include real-time program-interrupt signals and a complex input-output system. This input-output system, as well as communicating with over a dozen other digital data-handling devices, has more than 250 separate addresses. The discussion of the complex structure of the CDP program covers the three separate programs which run in an interwoven fashion. It describes the solution of problems caused by this interweaving and by the real-time program interrupt.

1335
Mathematical Circuit Analysis and Design (Sperry Rand Corp.); *U. S. Govt. Res. Rept.*, vol. 34, p. 579(A); November 18, 1960. PB 148 222 (order from LC mi$3.00, ph$6.30).

An attempt to develop a mechanized routine for the design and fault diagnosis of electronic circuits is described. The progress to date is summarized and areas for future study are suggested. Fault diagnosis has been considered initially, since: 1) it seems to be the more tractable of the two, and 2) circuit design is considered to be a direct extension of fault diagnosis, requiring the incorporation of the worst-case criterion and a method of optimization. A unique method has been devised for translating any schematic diagram into computer language so that each component and its location is specifically described. A method has been established and programmed for the Univac computer to generate the complete set of irredundant nodal and branch equations which will handle both linear and nonlinear characteristics of the circuit parameters. Finally, the set of equations has been reduced by eliminating all variables not required for solution of the component parameters. A sample circuit is given and a copy computer printout accompanies the description of each stage of the analysis.

1336
A Computer-Controlled Dynamic Servo Test System by V. A. Kaiser and J. L. Whittaker (Douglas Aircraft Co.); *Proc. Eastern Joint Computer Conf.*, pp. 255–266; December 13–15, 1960.

A computer-controlled dynamic servo-test system which carries out the operations normally performed manually in the testing of a missile control system is described. In obtaining the frequency response and stability characteristics of a control system, the computer operates to: 1) generate a prescribed sequence of driving functions, 2) sample the resulting outputs, 3) compute from these samples the gain and phase characteristics, and 4) provide tabulated or plotted results in a form ready for analysis. This application of a general-purpose digital computer will reduce vastly the time required in the development of missile control systems, subsystems, and components. The equipment used and the analytical techniques employed to enable automatic dynamic testing are described, and the time required and accuracy obtained using this new automatic facility are compared with those of the conventional manual method of servo testing.

Digital Computer Equipment for an Advanced Bombing Navigation and Missile Guidance Subsystem for the B-70 Air Vehicle—*see* 1457.

Flow-Table Logic—a Machine Organization for Microminiaturization—*see* 1323.

1337
PB 250, a High-Speed Serial General Purpose Computer Using Magneto-Strictive Delay Line Storage by R. M. Beck (Packard Bell Computer Corp.); *Proc. Eastern Joint Computer Conf.*, pp. 283–297; December 13–15, 1960.

The design objective of a general-purpose computer intended to serve as a component in special purpose systems is presented. Some of the design considerations applied toward meeting these objectives are also considered. The significance of the use of magnetostrictive delay memories in a low-cost computer as well as the approach used to minimize the active elements in a flexible computer input-output system are discussed.

1338
The RCA 601 System by A. T. Ling and K. Kozarsky (RCA); *Proc. Eastern Joint Computer Conf.*, pp. 173–177; December 13–15, 1960.

The design aspects of the RCA 601 system are discussed with particular emphasis on the three dimensions which may be readily modified: speed, function and capacity. The structural organization of the system and associated memory formats are described. The programming and system features which enable the system to utilize the fast memory and tapes, an arbitrary degree of simultaneity, and the generalized word and character structure are explored. Emphasis is placed on variable instruction length, address modification and listing structure. Design features, such as a 1.5-μsec memory, 120-kc tapes, and generalized word structure are also described.

1339
A Description of the IBM 7074 System by R. R. Bender, D. T. Doody, and P. N. Stoughton (IBM Corp.); *Proc. Eastern Joint Computer Conf.*, pp. 161–171; December 13–15, 1960.

The IBM 7074 system, the second major step in the IBM 7070 data-system family, provides increased processing power by improvements within the framework of the 7070. A new circuit card and the IBM standard modular system of packaging make possible system growth by substitution of functional units rather than by replacement of an entire system. Program compatibility with the 7070 is retained. Increased processing speeds are attained by faster circuits, full parallel arithmetic, and the use of faster storage. Circuits, packaging, and machine organization are described. Examples of instruction execution times are given, and their effects on system performance are discussed.

1340
The Instruction Unit of the STRETCH Computer by R. T. Blosk (IBM Corp.); *Proc. Eastern Joint Computer Conf.*, pp. 299–324; December 13–15, 1960.

The major functions of the instruction unit for the STRETCH computer are described, a general picture of the internal machine organization is given, and several examples of the achievement of performance goals are presented. The instruction unit, which is a large, complex, high-speed computer, is designed and built to provide the major function and control ability for the STRETCH computer. The instruction unit has a variety of functions; most important are the fetching and indexing of all instructions for the computer, and the execution of a large set of instructions dealing with index arithmetic, branching, and word transmission. The size and complexity of the unit are determined by the instruction buffering and the extensive amount of simultaneous operations required to achieve the high performance goals set for the computer.

1341
The Sumador Chino by J. L. Rogers (Burroughs Corp); *Commun. Assoc. Comp. Mach.*, vol. 3, pp. 621–622; November, 1960.

The Sumador Chino, or Chinese Adder, a mechanical device used in Mexico for adding, is described. The device consists of nine rotating cylinders on which addition tables are printed. Carries are indicated by means of subscripts. Unlike the abacus, the device cannot be used directly for subtraction, since for that operation the tables would have to be adjusted and borrow subscripts would replace the carry subscripts.

1342
CORSAIR, A Digital Differential Analyzer by P. L. Owen, M. F. Partridge, and T. R. H. Sizer (Royal Aircraft Estab.); *Electronic Engrg.*, vol. 32, pp. 740–745; December, 1960.

A digital differential analyzer known as CORSAIR is described. Differential analyzers operate by reducing all computation to the single operation of integration; variables may be integrated with respect to quantities other than time, and it is this facility which enables the computer to deal with nonlinear problems. Digital integration is performed by the summation of rectangles in a step-by-step process; electronic speeds of operation enable continuous quantities to be represented with great accuracy. Integrators are interconnected by transmitting pulses which represent increments of the computed quantity. In CORSAIR the state of each integrator is described by means of two numbers or words in binary code which are stored in a ferrite-core matrix. These number pairs are read out sequentially and the integration process takes place in a central time-sharing arithmetic unit.

1343
A Digital-Analog Pulse Amplitude Interpolation Computer by J. D. Schmidt, B. K. Eriksen, and F. H. Schlereth (General Electric Co.); *Proc. NEC*, vol. 16, pp. 600–610; 1960.

A computer circuit which automatically interpolates between two inputs of different amplitudes and adds or subtracts this quantity to a third input is described. This function is very useful in computing Doppler frequency in certain radar systems. A bipolar interpolation of the form $N_F = K(A/A+B)$ is performed, where N_F is the interpolated quantity, K an arbitrary constant, and A and B are the two inputs. The processing of the information is done by a hybrid digital-analog system with the interpolation being performed by an analog circuit and the addition or subtraction being performed digitally. The hybrid system gives an accuracy of 0.3 per cent in this application and results in a considerable saving in circuit complexity over a straight digital or analog system.

A New Active-Passive Network Simulator for Transient Field Problems—*see* 1415.

3. DEVICES AND BASIC LOGIC AND WAVEFORMING CIRCUITS

1344
Tunnel Diode Logic Circuits by R. H. Bergman (RCA); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-9, pp. 430–438; December, 1960.

The use of tunnel diodes in the logic and control portions of high-speed computers is discussed. Considerations of diode uniformity requirements, stability problems, and power supply requirements have led to a monostable type of logical circuit. The switching properties of this circuit are analyzed and found to depend upon the negative resistance-capacitance time constant of the unit. The basic function performed by the circuit is a thresholding operation from which a set of logical building blocks is derived. Compatible dynamic and bistable storage schemes are discussed. Of major importance is the effect of diode variations upon the logical gains and delays of the circuits. These properties have been tabulated for tunnel diodes with 5 per cent tolerances on knee current and voltage. Experimental circuits using diodes with a time constant of 1.4 nsec have given a nominal switching time of 7.5 nsec.

1345
Esaki Diode Logic Circuits by G. W. Neff, S. A. Butler, and D. L. Critchlow (IBM Corp.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-9, pp. 423–429; December, 1960.

Several digital-computer circuits which utilize the properties of tunnel diodes are described. In particular, shift registers, triggers, and counters are presented. The following shift registers are described: 1) A register which consists of one Esaki diode and one conventional diode per stage. Shifting is accomplished with a two-phase square-wave drive. The Esaki diode provides memory and power gain, and the conventional diode provides a unilateral flow of information. 2) A register which combines Esaki diodes with square-loop ferromagnetic cores. Again the Esaki diode provides memory and power gain. Upon application of a single-phase drive, the cores perform a gating operation depending upon the state of the diodes. 3) With the use of Esaki diode-transistor combinations, high-speed circuits are obtained which depend upon the Esaki diodes primarily for memory and the transistors for power gain and unilateral flow of information. The flip-flop and counter circuits presented are: 1) a binary counter using Esaki diodes with magnetic cores; 2) high-speed flip-flops using Esaki diode-transistor combinations.

1346
Calculated Waveforms for Tunnel Diode Locked Pair by H. R. Kaupp and D. R. Crosby (RCA); PROC. IRE, vol. 49, pp. 146–154, January, 1961; *Proc. Eastern Joint Computer Conf.*, pp. 233–240; December 13–15, 1960.

An introductory analysis of the tunnel-diode locked-pair circuit is presented. The characteristics of the tunnel diode, together with the simplicity of the locked-pair circuit, make it a major contender for use as a high-speed computer element. High speed and high gain are the main advantages of the locked pair; the three-phase power supply and lack of a simple means for logical inversion are its main disadvantages. The basic circuit consists of two tunnel diodes in series, the node common to the tunnel diodes being both the input and output terminal. As a computer element, the locked pair functions in much the same manner as the phase-locking harmonic oscillator (PLO). Like the PLO, the locked pair overcomes the difficulty of coincident input and output terminals by using a three-phase voltage source. The feasibility of using a digital computer to solve nonlinear circuit problems is also demonstrated. A digital computer makes possible an exact solution by doing away with relatively ineffectual linear approximation techniques. Furthermore, the stray parameters associated with laboratory work at high frequencies are excluded, thereby disclosing the true nature of the circuit.

1347
Current Build-Up in Avalanche Transistors with Resistance Loads by D. J. Hamilton (University of Arizona); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-9, pp. 456–460; December, 1960.

A transient analysis for the avalanche transistor is carried out through the use of a diffusion model described in terms of charge variables. Basically, the current is calculated as a function of time by taking the gradient of the minority carrier charge stored in the base region. Two methods of approximating the distribution of stored charge are described. Good agreement has been obtained between calculated and experimental results; it is found that the rise time for the resistance-load case is about four times that for a capacitance-load case which produces the same peak current. A practical pulse-generator circuit is described in which the resistance load takes the form of a delay line. The performance of this circuit is compared with that of a capacitance-load relaxation oscillator; while the rise time of the former is longer, the pulse shape is more easily controlled.

Transistorized Building Blocks for Data Instrumentation—*see* 1456.

1348
System Application of Hybrid Logic Circuitry by J. T. Lynch and J. J. Karew (Burroughs Corp.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-9, pp. 418–423; December, 1960.

A comparative performance rating of cir-

cuit techniques for performing logical functions in digital systems may be based upon: 1) reliability and simplicity, 2) input and output capabilities, 3) propagation time, and 4) cost. The "hybrid transistor diode logic" (HTDL) circuit technique employs either diodes or emitter-follower transistors as gates and buffers, to maximize the circuit performance rating. The HTDL technique thus combines the advantages of lumped and distributed gain circuits. The cascading of diodes and emitter followers in logical gate matrices can be analyzed as the transmission of binary signals through a video system of a given bandwidth. The HTDL technique optimizes the use of the transistor through nonsaturating, low-impedance circuitry. This optimum use of 200–500 Mc gain-bandwidth transistors is primarily limited by present-day packaging techniques and their inductive and capacitive loading effects upon the circuits. The development of macromodule packaging techniques, using 200–500-Mc transistors, in HTDL circuitry, would permit system speeds (synchronous clock rates) to exceed 50 Mc.

1349
Improvements to Current Switching by F. K. Buelow (IBM Corp.); IRE TRANS. ON COMPUTERS, vol. EC-9, pp. 415–418; December, 1960.

The use of diode switching circuits in conjunction with emitter followers and current switching circuits to evolve a new set of system building blocks is described. These blocks exhibit typical delays under five mμsec. Diodes cost less and are physically smaller than transistors; therefore, this new system is cheaper and faster than an all-current switching system and permits at least a fivefold increase in packaging density.

1350
A Computer Subsystem Using Kilomegacycle Subharmonic Oscillators by I. Abeyta, F. Borgini, and D. R. Crosby (RCA); PROC. IRE, vol. 49, pp. 128–135; January, 1961.

The problems associated with a high-speed carrier computer are discussed and a computer subsystem employing subharmonic oscillators driven with a 3.7-kMc sine-wave source is described. The three-phase power source was modulated at a 30-Mc rate so the interval between successive logic operations was 11 nsec. The logic module is a balanced resonant varactor circuit having a pulsed subharmonic output at 1.81 kMc and a rise time of 3 nsec. The output from each module, at a level of about 1 mw, is distributed through a network of resistors and coaxial lines to the modules of the succeeding power-supply phase. The logic is done at 1.85 kMc by forming a majority of three from the three inputs. This majority logic, with the readily available inversion operation, provides a universal set of logic gates. The subsystem consists of four main parts: the logic modules, the 1.85-kMc signal distribution, the 3.7-kMc power distribution, and the power supply. The subharmonic oscillator has advantages characteristic of a carrier computer; among these is the ability to transform impedance levels and the ability to get a logic inversion by adding a cable length to give 180° phase change of the signal. Advantages peculiar to the subharmonic oscillator system are the simplicity of the varactor and its inherently fast operation. This approach also has a high logic gain, and provides fully timed and amplitude-limited pulses at the output of each module. Three-phase operation gives an effective directivity to the single-port gate. The subsystem contains a three-phase ring, two majority gates, and their majority logic drivers, comprising a total of seven subharmonic oscillators. Operation is stable, with reasonable allowable variations in the power-supply frequency and amplitude.

The Simulation of Neural Elements by Electrical Networks Based on Multi-Aperture Magnetic Cores—*see* 1391.

1351
Transient Analysis of Cryotron Networks by Computer Simulation by M. K. Haynes (IBM Corp.); PROC. IRE, vol. 49, pp. 245–257; January, 1961.

A general method is derived for transient analysis of complicated nonlinear dynamical systems by use of a digital computer programmed to perform tensor transformations and numerical integration. Tensor methods, adapted from Kron's techniques, are used for converting circuit data into a form for transient simulation by numerical integration. An IBM 704 program has been written for simulation of cryotron networks. This simulator has been used to study switching speeds of cross-latched cryotron flip-flops, five-stage free-running ring circuits, and a three-bit, self-timing, self-checking binary parallel adder. The adder circuit contains 233 circuit elements, including 93 cryotrons arranged in 55 meshes. Results of these studies are included.

1352
A Secondary-Emission Pulse Circuit, Its Analysis and Application by J. A. Narud (IBM Corp.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-9, pp. 439–451; December, 1960.

A regenerative pulse circuit using a single secondary-emission tube that is able to generate pulses having a rise time of 6 mμsec and a width continuously variable from 25 mμsec to 12 μsec is described. First a theoretical discussion of the circuit is given in which expressions for pulse width and rise time are derived. Then various practical realizations of the circuit are presented. Among others, these include a mμsec pulse generator and a fast pulse-height discriminator.

Avalanche Transistor Pulse Generator—*see* 1347.

1353
A Binary Counter and Accumulator Using Transistors and Magnetic Cores by H. R. Irons (Naval Ord. Lab.); *U. S. Govt. Res. Repts.*, vol. 34, p. 576(A); November 18, 1960. PB 147 565 (order from LC mi$2.70, ph$4.80).

Binary components which employ a transistor-magnetic-core combination as a fundamental building block for digital circuits are described. These components are capable of high operating speeds and yet require very little power at low operating speeds. For example, a five-stage binary counter has a total power consumption of 40 μw when operated at a counting rate of 100 pps. The same counter is capable of operation at rates as high as 2×10^6 pps. Good reliability is indicated by the fact that a counter has been operated successfully over a temperature range of −65°F to 160°F with a supply voltage of −2.0 volts to −4.5 volts.

1354
Decoding Nets and the Theory of Graphs by Z. Pawlak (Warsaw Polytech. Inst.); *J. Soc. Indust. Appl. Math.*, vol. 7, pp. 1–5; March, 1959.

The problem of minimizing decoding nets (which are basic components of digital computers) is expressed in terms of linear graphs, with diodes interpreted as branches and wire connections between diodes as nodes of a graph. A necessary condition for a decoding net to have a minimal number of diodes is given.

4. STORAGE AND INPUT-OUTPUT

1355
Computer Memories. A Survey of the State-of-the-Art by J. A. Rajchman (RCA Labs.); PROC. IRE, vol. 49, pp. 104–127; January, 1961.

Computer-memory developments of the last decade, the present state, and efforts for improvements are surveyed. The following topics are included: principles of storage and selection of random-access memories; principles and engineering considerations of current-coincident-driven core memories; magnetic decoding and load-sharing switches; word-organized one-core and two-core-per-bit memories; fast and impulse switching; transfluxor memories; nondestructive read-out memories; ferrite-apertured plates; twistors; fixed read-only memories; thin magnetic-film memories—dots, sheets, coated wires and rods; present operational memories typically with capacities of 10^5 to 10^6 bits and read-write cycles of 2 to 15 μsec; likelihood of the order of 100-nsec read-write cycle times attainable with ferrite and thin film memories; consideration relating to large capacities; ferroelectric memories, cryoelectric superconductive memories, superconductive films, Crowe cells, continuous sheets, systems, and the outlook for large capacities; tunnel-diode memories which promise a read-write cycle of the order of 10 nsec; and outlook for content addressable memories.

1356
Ferrite-Core Memory Systems with Rapid Cycle Times by D. B. G. Edwards, M. J. Lanigan, and T. Kilburn (Manchester University); *Proc. IEE*, vol. 107B, pp. 585–598; November, 1960.

Improvements in storage systems using currently available square-loop ferrite cores are considered. These enable the normal cycle time of 6–10 μsec to be reduced to less than 2 μsec. Effort has been concentrated on the word-selected two-core-per-digit arrangement, and the most promising techniques are those which involve partial-flux switching. A system suitable for a store of

1024 words of 52 digits with a cycle time of about 1.6 μsec is developed. In a smaller store of 100 words, a cycle time of approximately 0.6 μsec is feasible.

1357
A Digital Computer Store with Very Short Read Time by T. Kilburn and R. D. Grimsdale (University of Manchester); *Proc. IEE*, vol. 107B, pp. 567–572; November, 1960.

The operating principles and the construction of storage units with a very short read access time are described. One form of the store which has been constructed has a capacity of 200,000 bits of permanent information, and another has been built with a capacity of 100,000 bits, the whole contents of which may be changed in under one minute. The technique employed permits the construction of very large stores at low cost. Each digit cell consists basically of two sets of windings which form the primary and secondary of a transformer. The two binary states are determined by the presence or absence of a piece of linear ferrite material coupling the windings. The access time is largely determined by the physical size of the store and the speed of operation of the associated circuits; a time of 100 mμsec is typical.

1358
Magnetic Film Memory Design by J. I. Raffel, T. S. Crowther, A. H. Anderson, and T. O. Herndon (M.I.T. Lincoln Lab.); PROC. IRE, vol. 49, pp. 155–164; January, 1961.

Thin magnetic films of permalloy have characteristics ideal for high-speed digital storage. A simple rotational model modified to include the effects of wall switching and dispersion of the preferred direction of magnetization provides a basis for describing properties of engineering interest. A selection system has been chosen which allows great latitude in film uniformity. Production of films with magnetic properties uniform to within ±10 per cent is readily achieved. Specifications for operation in a destructive mode can easily be met by existing film arrays; the nondestructive mode is considerably more stringent unless very small signals can be tolerated. The first film memory, which has been in reliable operation since the summer of 1959, has 32 ten-bit words and has been operated with a minimum cycle time of 0.4 μsec. Higher speed and larger capacities will require higher bit densities and improved techniques to minimize undesirable coupling between drive and sense lines. The use of 10×60-mil rectangles, balanced sense windings, and longer words, will hopefully permit memories of about 200,000 bits with cycle time under 0.2 μsec.

1359
High Speed Thin Magnetic Film Memories by R. Bogusch and E. A. Fisch (General Electric Co.); *Proc. NEC*, vol. 16, pp. 840–845; 1960.

The development of high-speed thin magnetic film memories is described. A brief description of the properties of the film is given, along with a discussion of the phenomenon of rotational switching. The transistorized driving and sensing circuitry is also described.

1360
A Computer Storage Matrix Using Ferromagnetic Thin Films by E. M. Bradley (Int. Computers and Tabulators Ltd.); *J. Brit. IRE*, vol. 20, pp. 765–784; October, 1960.

A new method for selecting a magnetic-film storage element from a matrix of elements which exploits the coherent rotational mode of reversal is described. It is shown that, by using a film made of a newly developed alloy "Gyralloy 1," the reproducibility problems previously encountered can be solved. It is shown that the use of aluminum as substrate for the film enables the signal-noise ratio to be increased to an acceptable level even for very large stores; the drive power requirements are shown to correspondingly reduced. Details of performance and construction of a storage device which contains 50 words each of 50 bits are given.

1361
An Electrically Alterable Nondestructive Twistor Memory by R. L. Gray (Burroughs Corp.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-9, pp. 451–455; December, 1960.

The basic principles of twistor operation are discussed and it is shown how the twistor may be fabricated into a memory. A nondestructive method of reading a twistor memory by the use of multiple solenoids is described. A typical configuration of a twistor memory which, by the use of this nondestructive reading method, may be operated either in a destructive mode or in a nondestructive mode, is presented.

Associative Self-Sorting Memory—*see* 1380.
Computer Using Magnetostrictive Delay-Line Storage—*see* 1337.

1362
The Printed Motor: A New Approach to Intermittent and Continuous Motion Devices in Data Processing Equipment by R. P. Burr (Circuit Res. Co.); *Proc. Eastern Joint Computer Conf.*, pp. 325–342; December 13–15, 1960.

The printed dc motor is characterized by high pulse torque capability and freedom from cogging or preferred armature positions. These attributes lead to a variety of applications in data-processing equipment ranging from reel and capstan drives in magnetic and paper tape transports through detenting and positioning mechanisms. Analysis of the motor on a velocity basis yields a simple equivalent circuit which is a powerful tool for designing both the machine and its drive circuits into a specific requirement. Since there is no rotating iron in the structure and since the field is supplied by permanent magnets, the speed-torque curve of the motor is a straight line whose slope defines a "mechanical source impedance." Inertia of the proposed load appears as a capacitor in the same dimensional system. When the desired machine motion can be expressed in terms of velocity and the inertia of the load is known, the shape and magnitude of the necessary driving signal together with the power which must be expended can all be determined for the operating cycle. A typical example of an application in a paper tape transport is described.

1363
The Development of the Flexible-Disk Magnetic Recorder by R. T. Pearson (Lab. for Electronics, Inc.); PROC. IRE, vol. 49, pp. 164–174; January, 1961.

A brief history of the development of the flexible-disk magnetic recorder is presented. Principles of the aeroelastic behavior of the disk and the results of a mathematical analysis of the equilibrium operating conditions are discussed. Experimental results which illustrate the effects of operating parameters on disk dynamics are given. These results indicate the design considerations necessary to produce a wide range of new storage devices. The characteristics of some of the newly developed models are presented.

1364
Univac-RandexII-Random Access Data Storage System by G. J. Axel (Sperry Rand Corp.); *Proc. Eastern Joint Computer Conf.*, pp. 189–204; December 13–15, 1960.

A random-access drum file having 198.6 million bits total storage capacity, a bit density of 650 pulses per inch, and 385-msec average data-access time is described. Two flying-magnetic-recording heads transfer data to and from the drum file unit. They are self-supported, by a hydrodynamically-generated air film, over two magnetically-plated drums (24 inches diameter and 44 inches long). The heads, drums, and head-positioning servo are enclosed in a sealed and pressurized chamber to prevent their contamination by foreign material normally found in the atmospheric air. The logic of operation of the over-all drum file, construction of the flying heads, and the servo and mechanical adder which positions the flying heads over selected addresses on the drum are described in detail.

High Speed Data Transmission Systems—*see* 1435.

1365
Hot-Wire Anemometer Paper Tape Reader by J. H. Jory (Soroban Engrg., Inc.); *Proc. Eastern Joint Computer Conf.*, pp. 267–268; December 13–15, 1960.

A hot-wire anemometer-type reader is proposed as a method of achieving reliable, high-speed reading of perforated paper tape. The principle of operation concerns the change in resistance of fine coil of wire of known temperature coefficient of resistance when subjected to an air stream directed through a perforation in a paper tape bebeing read.

1366
Computer Generated Displays by R. T. Loewe, R. L. Sisson, and P. Horowitz (Ford Motor Co.); PROC. IRE, vol. 49, pp. 185–195; January, 1961.

The need for effective computer-generated displays of symbols and lines in various formats which results from an increased use of computer output for decision-making is discussed. Two key functions involved in display generation are digital-to-image conversion, and image storage for projection. Symbol generation can be performed by use of logical and electron beam devices to create dot patterns, intensity-modulated scans,

mixtures of waveforms or shaped electron beams. Lines are generated as dots, line segments, or vectors. Large, bright displays are obtained by optically projecting images using CRT, photographic, electrostatic, oil film, or thermoplastic techniques. Many new developments show promise for application in computer-generated displays.

1367
Asynchronous Circuits for Entering Data into An Alphanumeric Visual Display System by E. L. Younker (Stanford Res. Inst.); *Proc. NEC*, vol. 16, pp. 823–828; 1960.

A visual display system combined with a transistor-driven magnetic-core memory to provide a flexible means of storing and displaying numerical and alphabetic information is described. Storage is provided for 190 characters which are displayed on a conventional cathode-ray tube (*e.g.*, a TV picture tube) at a frame rate of sixty per second. Information can be inserted into the core memory manually by means of electromechanical keyboards and automatically from a magnetic tape. The design of control circuits for entering asynchronously-occurring information into the clock-controlled visual display system is discussed. Two cases of the asynchronous-synchronous problem are considered: 1) where the rate of data input is lower than the rate of acceptance by the display, and 2) where the rate of input data is higher than the rate of acceptance by the display. The first is illustrated by the entry of data from a manually-operated keyboard; the second by the entry of data from a magnetic tape. The asynchronous control circuits have been operated more than 1000 hours and have shown a high degree of reliability in handling incoming data.

5. PROGRAMMING AND CODING

1368
Some Applications of Logical Syntax to Digital Computer Programming by R. M. Carp (Harvard University); *U. S. Govt. Res. Repts.*, vol. 34, p. 567(A); November 18, 1960. PB 147 350 (order from LC mi$8.70, ph$30.30).

The methods of logical syntax are applied to the analysis of programming languages. The programming languages considered include not only computer languages, but also flow-charting languages and the source languages of automatic programming systems. Syntactic models are defined and used for the development of analytic checkout procedures and techniques for improving the efficiency of programs.

1369
TABSOL a Fundamental Concept for Systems Oriented Languages by T. F. Kavanagh (General Electric Co.); *Proc. Eastern Joint Computer Conf.*, pp. 117–136; December 13–15, 1960.

The use of decision structure tables to describe complex, sequential, multi-variable, multi-result decision systems is considered. TABSOL, an automatic programming technique for solving structure tables on any computer, is also discussed. The structure table and TABSOL concepts are major steps forward in describing complex operating decision systems since they replace both flow charting and computer coding. In addition, changes can be readily introduced by the systems designer, greatly simplifying the systems maintenance problem. By forcing a logical step-by-step analysis, these techniques highlight business causal relationships and simplify debugging in the systems designer's own language.

1370
Survey of Coded Character Representation by R. W. Bemer (IBM Corp.); *Commun. Assoc. Comp. Mach.*, vol. 3, pp. 639–641; December, 1960.

A chart showing the logical representation of character sets in paper tape, magnetic tape, and the main storage of a number of different machine systems is presented. Sixty-four character modules are used, and the positions are designated in the octal number system from 00 to 77. The chart is published to show the need for standardization in the data-processing industry and to obtain further information from experts.

1371
A Comparison of 650 Programming Methods by T. B. Curtz, J. F. Riordan, and M. Spohn (University of Michigan); *Commun. Assoc. Comp. Mach.*, vol. 3, pp. 663–664, 671; December, 1960.

Two IBM 650 programming compiling systems, GAT, prepared by the University of Michigan, and RUNCIBLE, developed by the Case Institute of Technology, are critically evaluated and compared. Each compiler was used with the SOAP assembly program. A typical mix of scientific problems was run in both systems. It is concluded that the use of both compilers should be confined to the amateur programmer operating in an open shop, since neither system is very efficient. GAT is considered to be more convenient for a machine having floating point, index registers, and special character facilities, since it is better documented and handles input-output, standard mathematical notation, and subroutines more succinctly and flexibly.

1372
DYANA—A Computer Program for the Automatic Analysis of Dynamic Systems by D. E. Hart and B. Hargreaves (General Motors Res. Labs.); *Proc. NEC*, vol. 16, pp. 308–315; 1960.

DYANA is a compiler for the IBM 704 computer that can be used to solve vibrational, electrical, and equivalent dynamics problems. The DYANA language, a problem-oriented language, is used by the engineer to describe his vibrational or electrical system. The DYANA compiler analyzes the system description, prepares a methematical model for the system, determines which numerical techniques are required to solve the mathematical model, and codes a FORTRAN program that can be used by the engineer in the study of his system. The DYANA system is thus able to transfer from the engineer to the computer many of the routine functions that are not commensurate with an engineering education.

1373
An Assembly Program for a Phase Structure Language by R. A. Brooker and D. Morris (University of Manchester); *Computer J.*, vol. 3, pp. 168–174; October, 1960.

The structure of an assembly program which allows the user to define the meaning of statements he uses is described. These statements are not restricted to those of a procedural or imperative nature, but may also be declarative. The application is directed at the Ferranti Atlas Computer, and it is hoped that the approach advocated will considerably reduce the time required to write autocodes.

1374
Compilation for Two Computers with NELIAC by K. S. Masterson, Jr. (U. S. Naval Postgraduate School); *Commun. Assoc. Comp. Mach.*, vol. 3, pp. 607–611; November, 1960.

A compiler based on ALGOL and named NELIAC, developed as a "bootstrap" compiler for the Remington Rand COUNTESS computer, is described. The compiler was first used to produce a version of itself which, running as a COUNTESS program, generated machine code for a compiler for the CDC 1604 computer. This second compiler was then used to generate a version of NELIAC for the CDC 1604. All three versions of NELIAC can accept virtually identical input language.

1375
Automatic Coding for Business Applications by R. M. Paine (Internatl. Computers and Tabulators Ltd.); *Computer J.*, vol. 3, pp. 144–149; October, 1960.

The aims of automatic programming are set forth and its constituent elements are enumerated. Its application to business data problems is considered and some of the advantages and disadvantages are discussed. The goal of a common program-oriented language is set forth and a brief description of COBOL is supplied.

1376
Over-All Computation Control and Labelling by A. Holt (University of Pennsylvania); *Commun. Assoc. Comp. Mach.*, vol. 3, pp. 614–615; November, 1960.

The tasks which should be expected of a general control program supervising a group of service programs are discussed. Stress is laid on the importance of labelling all computer programs and input and output tapes comprehensively. In this manner, operator actions may be standardized and the risk of errors minimized.

Multiple Processing Bit-by-Bit—*see* 1324.

1377
An Estimation of the Relative Efficiency of Two Internal Sorting Methods by H. Nagler (IBM Corp.); *Commun. Assoc. Comp. Mach.*, vol. 3, pp. 618–620; November, 1960.

Two internal sorting methods, the conventional two-way merge and a form of binary search, are compared. Formulas for the number of computer steps required in each method as functions of the number of characters in the key and the length of sequence produced are derived. A table determining for the IBM 705 at what sequence length the binary search should give way to the two-way merge is presented.

1378
Polyphase Merge Sorting—An Advanced Technique by R. L. Gilstad (Minneapolis-Honeywell Regulator Co.); *Proc. Eastern Joint Computer Conf.*, pp. 143–148; December 13–15, 1960.

New merge-sorting techniques that utilize tape drives more efficiently than conventional sorting methods are described. The Cascade sorting method is reviewed as background for a new advance, called polyphase sorting. The methods used in polyphase sorting are explained in terms recognizable to anyone familiar with merge sorting on computers. The merging powers of normal merge sorting, Cascade, and polyphase techniques are compared, and it is concluded that Cascade sorting and polyphase sorting represent techniques which will make the new generation of computers even more powerful than before in one of the most common areas of computer usage.

1379
Computer Time for Address Calculation Sorting by I. Flores (Norwalk, Conn.); *J. Assoc. Comp. Mach.*, vol. 7, pp. 389–409; October, 1960.

A method is developed of sorting by address calculation, by which a location in a file is determined by a linear formula. If the location is already full, a search for the nearest empty location consonant with preservation of the order of the items is made. The item at the specified location, together with adjacent items, is moved in a systematic manner. Best results are obtained for file capacity 50 per cent to 150 per cent larger than the number of items to be sorted. Estimates for the sorting time, including the final packing of items and the proportion of misplaced items, are derived.

1380
Associative Self-Sorting Memory by R. R. Seeber, Jr. (IBM Corp); *Proc. Eastern Joint Computer Conf.*, pp. 179–187; December 13–15; 1960.

A memory system which automatically performs the function of sorting data is proposed. An associative memory, based on cryotron circuits, is extended to permit high, equal, or low comparison of the interrogating word with all words in memory. This comparison permits the new word to be placed between the proper pair of sorted words in the memory. A double shifting operation is used to move the appropriate block of words to make room for the new word.

1381
Fibonaccian Searching by D. E. Ferguson (Computer Sciences Corp.); *Commun. Assoc. Comp. Mach.*, vol. 3, p. 648; December, 1960.

A variant of the well-known binary search technique for searching an ordered list which makes use of Fibonacci number is described. The advantage claimed is that addresses used for successive trials may be calculated by subtraction rather than by division as required by a machine not equipped with a binary shift. If at some iteration the item has been isolated to an interval of size u_i (where u_i is the ith Fibonacci number), then at the following iteration the item will be isolated to an interval of size u_{i-1} or u_{i-2}. The expected searching time is of order $\log_2 n$ and the maximum searching time is of order $\log_\phi n$ where $\phi=(1+\sqrt{5})/2$.

1382
Theory of Files by L. Lombardi (University of California); *Proc. Eastern Joint Computer Conf.*, pp. 137–141; December 13–15, 1960.

The theory of files is discussed as a tool for the logico-mathematical treatment of automatic non-numerical data-processing problems, such as machine accounting, information retrieval, and mechanical translation of languages. The main result which has been obtained from the application of this theory is the recent discovery of a simple pattern to which the data flow of any information-processing procedure conforms, regardless of how many files are involved. The flow of each file can be controlled and coordinated with the flow of the others by means of five Boolean parameters, called indicators. A specially-designed algebraic business language (ABL) exploits these results for the purpose of programming digital data processors. The impact of the theory of files upon the logical design of digital systems is also considered. A system language called ABL which allows the use of mathematical methods for the description of non-numerical information processing is described. Files and sets of files are defined and discussed mathematically. The nature of the laws which coordinate the flow of data of non-numerical systems is investigated. These laws, which appear to be common to all non-numerical procedures, can be stated by means of certain Boolean variables.

General Purpose Programming for Business Applications—*see* 1458.

6. FORMAL AND NATURAL LANGUAGES, INFORMATION RETRIEVAL, AND HUMANITIES

1383
The Present Status of Automatic Translation of Language by Y. Bar-Hillel (Hebrew University, Jerusalem); *Advances in Computers*, Academic Press, Inc., New York, N. Y., vol. 1, pp. 91–163; 1960.

The aims of machine translation of languages are critically appraised. It is argued that fully-automatic high-quality translation is a presently unattainable goal, although an economic partly-mechanized translation process ought to be realizable in the near future. The work of all the important translation groups in the U. S., Europe, Russia, and Japan is critically surveyed. An extensive list of references is provided.

7. BEHAVIORAL SCIENCE AND ARTIFICIAL INTELLIGENCE

1384
Steps Toward Artificial Intelligence by M. Minsky (M.I.T. Electronics Res. Lab.); PROC. IRE, vol. 49, pp. 8–30; January, 1961.

The problems of heuristic programming—of making computers solve really difficult problems—are divided into five main areas: search, pattern-recognition, learning, planning, and induction. A computer can do, in a sense, only what it is told to do. But even when we do not know how to solve a certain problem, we may program a machine (computer) to *search* through some large space of solution attempts. Unfortunately, this usually leads to an enormously inefficient process. With *pattern-recognition* techniques, efficiency can often be improved by restricting the application of the machine's methods to appropriate problems. Pattern-recognition, together with *learning*, can be used to exploit generalizations based on accumulated experience, further reducing search. By analyzing the situation, using *planning* methods, we may obtain a fundamental improvement by replacing the given search with a much smaller, more appropriate exploration. To manage broad classes of problems, machines will need to construct models of their environments, using some scheme for *induction*. Wherever appropriate, the discussion is supported by extensive citation of the literature and by descriptions of a few of the most successful heuristic (problem-solving) programs constructed to date.

1385
FILTER—A Topological Pattern Separation Computer Program by D. J. Innes (Lawrence Radiation Lab.); *Proc. Eastern Joint Computer Conf.*, pp. 25–37; December 13–15, 1960.

A scanning technique which allows rapid recognition, separation, and measurement of the photographic records of star-type nuclear events in a bubble chamber is described. A device known as the spiral reader measures background and star-type event features, impartially discriminating against nonradial patterns by the geometry of its rotating scanning element. The event measurements are separated from the background measurements by an IBM 709 computer under the direction of a program called FILTER. The separated nuclear event measurements are subsequently reconstructed in space for physics analysis. FILTER exploits the observation that if a segment of a circular arc is rotated about a point on that arc, intercepts occur at regular intervals along a radius to the point at constant angular intervals of the rotation azimuth. The spiral reader, by placing the burden of event discrimination on a high-speed digital computer, minimizes the need for either special analysis equipment or for a human operator to make the topological separation. Simulation, calibration and cathode-ray tube display routines have been included in the FILTER system.

1386
Statistical Recognition Functions and the Design of Pattern Recognizers by T. Marill and D. M. Green (Bolt, Beranek and Newman Inc.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-9, pp. 472–477; December, 1960.

According to the model discussed in this paper, a pattern recognizer is said to consist of two parts: a receptor, which generates a set of measurements of the physical sample to be recognized, and a categorizer, which assigns each set of measurements to one of a finite number of categories. The rule of operation of the categorizer is called the "recognition function." The optimization of the recognition function is discussed, and the form of the optimal function is derived. In practice, a prohibitively large sample is required to provide a basis for estimating the

optimal recognition function. If, however, certain assumptions about the probability distributions of the measurements are warranted, recognition functions that are asymptotically optimal may be obtained readily. A small numerical example, involving the recognition of the hand-printed characters *A*, *B*, and *C*, is solved by means of the techniques described. The recognition accuracy is found to be 95 per cent.

1387
Pattern Recognition Using Autocorrelation by L. P. Horwitz and G. L. Shelton, Jr. (IBM Res. Ctr.); PROC. IRE, vol. 49, pp. 175–185; January, 1961.

A class of techniques for character recognition is described. These techniques are characterized by the property that the only parameters of the input that are used are those which are independent of the position of the character; that is, these techniques are registration invariant. The registration-invariant mathematical formalism which underlies these techniques is discussed, and the physical realization of several recognition schemes based on these concepts is described. Some results from a computer simulation of these procedures are given.

1388
Machine Recognition of Spoken Words by R. Fatehchand (Birbeck College, London); *Advances in Computers*, Academic Press, Inc., New York, N. Y., vol. 1, pp. 193–229; 1960.

The acoustics of the speech process and methods of machine analysis of basic speech sounds are reviewed. Machines exist which will recognize accurately a limited number of words such as digits, but storage requirements prohibit the expansion of this technique to the recognition of everyday speech on a word basis. The only existing technique for a recognition procedure based on phonemes is a binary choice method, and no attempt has been made to analyze continuous speech into discrete sounds. The speech recognition problem is considerably akin to that of two-dimensional pattern recognition. Any advance in one of these fields will influence the other.

1389
A Method of Voice Communication with a Digital Computer by S. R. Petrick and H. M. Willett (AF Cambridge Research Lab.); *Proc. Eastern Joint Computer Conf.*, pp. 11–24; December 13–15, 1960.

A pattern-recognition procedure for achieving automatic recognition of spoken using an eighteen-channel vocoder and a general-purpose medium-scale computer is described. If the speaker's own voice is used to prepare "masks" of the words he wishes to be recognized, correct identifications are made and printed on the computer flexowriter with almost 100 per cent accuracy. Arbitrary new spoken input words may be added in real time to the computer vocabulary. Other programs dependent upon this word-recognition facility which have been written include: 1) An interpretive routine which enables a speaker to say a sequence of words (from the set, zero, one, · · · , nine, plus, minus, times, bracket, equals) which are followed by a print out of the words spoken and the value of the expression defined. 2) A speaker recognition program which identifies the talker with appropriate comments as well as the word he spoke. 3) An adaptive program which enables the computer to reorient itself automatically to a new speaker's voice.

1390
Human Decision Making as Related to Air Surveillance Systems by A. P. Chenzoff and R. L. Crittenden (Dunlap and Assoc.); *U. S. Govt. Res. Repts.*, vol. 34, p. 610(A); November 18, 1960. PB 149 215 (order from LC mi$6.30, ph$19.80).

The literature on decision making is reviewed. The objectives of the survey were: 1) to describe some of the decision-making information now available in the areas of mathematics, experimental psychology, sociology and systems research; 2) to identify areas for further experimentation on human decision making. For many years to come, human decision making will remain of great practical importance for many kinds of decisions even in those situations where machines are being used to the fullest extent. These experiments therefore consider as far as possible those aspects of decision making that humans must undertake in air surveillance.

1391
The Simulation of Neural Elements by Electrical Networks Based on Multi-Aperture Magnetic Cores by A. E. Brain (Stanford Res. Inst.); PROC. IRE, vol. 49, pp. 49–52; January, 1961.

Multilevel storage, gating, controlling threshold level, and summation are basic functional operations common to several systems for the simulation of neural behavior by electrical networks. It is shown how these may be realized very efficiently in terms of multi-aperture magnetic cores; in a typical example, only two cores of the type described by Crane are required for a circuit with threshold control of three simultaneous gates and nondestructive analog read out. A logic element equivalent to a self-holding relay exhibits dc current gain and has compatible input-output connections.

1392
Plastic Neurons as Memory Elements by D. G. Willis (Lockheed Aircraft Corp.); *Proc. Internatl. Conf. on Information Processing*, UNESCO, Paris, pp. 290–298, June 15–20, 1959; *U. S. Govt. Res. Rept.*, vol. 34, p. 318(A), September 16, 1960. PB 147 670 (order from LC mi$2.70, ph$4.80.).

A class of plastic neuron models is defined, their information storage capacity is determined, and the mechanism by which information can be read into and out of a system of such neurons on a random-access basis is shown. The neuron model investigated has an arbitrary number, n, of input leads and one output lead. At any time step, the ith input lead will have associated with it some "synaptic" value s_i, and will be carrying some signal e_i, which is either 1 or 0 (firing or not firing). The neuron will fire and produce a 1 signal on its output lead if, at the previous time step, the sum of the n products $e_i s_i$ was greater than or equal to some threshold value T. The synaptic value s_i will increase, decrease, or remain unchanged at each time step, depending upon the firing condition of the neuron and the signal just carried by the ith lead. The storage capacity of this neuron model is determined principally by the long-term stability of the s values, and for any reasonable number of inputs, appears to be compatible with the requirements for human memory. Two demonstrations showing the read-in-read-out properties of a system of 288 plastic neurons simulated on a digital computer are presented, and an expression for the output of a plastic neuron as a function of its previous history is derived.

1393
Self-Organizing Systems—A Review and Commentary by J. K. Hawkins (Ford Motor Co.); PROC. IRE, vol. 49, pp. 31–48; January, 1961.

The class of self-organizing systems represented by networks which learn to recognize patterns is reviewed from an historical standpoint, and some of the behavioral similarities between such nets and biological nervous systems are discussed. Examples and results of several experimental models for alphanumeric character recognition are presented. The network synthesis problem is then recast in terms of redundant-information removal, multivariable curve-fitting, and expansion in orthonormal functions. Recognition network structures and the learning process are described from these points of view. The potential component and behavioral advantages to be gained from sequential feedback networks are discussed briefly.

1394
A Method for Overlapping and Erasure of Lists by G. E. Collins (IBM Corp.); *Commun. Assoc. Comp. Mach.*, vol. 3, pp. 655–657; December, 1960.

A method of overlapping in the computer storage of lists having data with multiple occurrence is described. The problem of erasing lists no longer needed is met by the arbitrary interspersion in the lists of words containing data counts. If an item in a list is borrowed by another list, the data count of the borrowed list is increased by one. Similarly, if a list is erased, all the data counts of items borrowed by it are diminished by one. In this manner efficient use is made of storage space by not duplicating multiply-used items.

1395
Redundancy Exploitation in the Computer Solution of Double-Crostics by E. S. Spiegelthal (General Electric Co.); *Proc. Eastern Joint Computer Conf.*, pp. 39–56; December 13–15, 1960.

There are many data-processing applications for which exact algorithmic processing schemes are either not strictly required or defy precise specification, or both. Such applications as machine translation, automatic abstracting and automatic indexing fall in this category. The human beings who execute these tasks make heavy use of the redundancy of the input data. What is required for their automation is some heuristic scheme for taking advantage of this redundancy. One such scheme is described. This

scheme, in its first concrete realization, has been used to solve double-crostic puzzles. Both the 704 programs for double-crostic solution and the general heuristic scheme are discussed.

1396
Programming Computers to Play Games by A. L. Samuel (IBM Corp.); *Advances in Computers*, Academic Press, Inc., New York, N. Y., vol. 1, pp. 165–192; 1960.

The history and characteristics of various attempts to program computers to play games, with the emphasis on chess and checkers, are surveyed. The concepts of degree of look-ahead and a scoring polynomial whose value the program seeks to increase are described. Two basic learning techniques, rote learning and learning by generalization, are discussed. Rote-learning procedures are usually more effective for chess beginning and end games, while learning by generalization is more effective for middle games. The importance of programming games is that experience is gained in models which are sufficiently simple as to be manageable, yet provide reasonable facsimiles of more complicated mechanisms and situations.

8. MATHEMATICS

1397
Uniqueness of Weighted Code Representations by G. P. Weeg (Michigan State University); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-9, pp. 487–489; December, 1960.

Decimal computers ordinarily use a binary-coded decimal representation. One class of binary-coded decimal digits is the so-called four-bit weighted code representation with weights w_1, w_2, w_3, w_4. Each w_i is a nonzero integer in the range $-9 \leq w_i \leq 9$, and the set of weights must have the property that every decimal digit can be represented by the sum

$$\sum_{i=1}^{4} b_i w_i,$$

with the b_i being 0 or 1. For some weighted codes the weights are such that some digits can be represented by more than one sum of the specified form. For example, the 7421 weighted code has the property that 7 may be represented either as 1000 or as 0111. This paper produces a necessary and sufficient condition on the weights of a weighted code for the unique representation of each digit by a sum of the specified form. Further, all possible sets of weights are displayed.

1398
Polynomials Orthogonal over Discrete Domains by B. A. Dent (Burroughs Corp.) and A. Newhouse (University of Houston); *SIAM Rev.*, vol. 1, pp. 15–59; January, 1959.

A simple method for deriving the coefficients of a set of orthogonal polynomials, to be used with least-squares curve-fitting of nonequidistant data, is presented. The results can be applied to previous work by Guest to obtain the probable error of the approximation. A detailed example for the case of three points is given.

1399
Orthogonal Functions Whose *K*th Derivatives are also Orthogonal by F. M. Stein (Colorado State University); *SIAM Rev.*, vol. 1, pp. 167–170; July, 1959.

The following result is established: the Hermite, Jacobi, and Laguerre polynomials form sets of functions which are orthogonal with respect to appropriate weight functions and over appropriate intervals, and their respective derivatives form sets of functions orthogonal over the same intervals as before (with perhaps different weight functions). Moreover, these polynomials are the only ones with this property.

1400
Least Squares Fitting of a Great Circle Through Points on a Sphere by L. de Witte (Hughes Aircraft Co.); *Commun. Assoc. Comp. Mach.*, vol. 3, pp. 611–613; November, 1960.

A method for fitting a great circle in the least-squares sense to a number of points lying approximately on a great circle is described. The spherical trigonometry required to express the problem is presented.

1401
Smooth Interpolation by T. J. Rivlin (Fairchild Engine Div.); *SIAM Rev.*, vol. 1, pp. 60–63; January, 1959.

The interpolatory polynomial is a unique polynomial of degree not greater than $n-1$ which interpolates a given set of n points. The oscillatory behavior of this polynomial decreases its usefulness in certain applications such as numerical differentiation. A method of interpolating n points by an nth-degree polynomial which results in a less oscillatory representation is presented. An example is given for the case of four points.

A Digital-Analog Pulse Amplitude Interpolation Computer—*see* 1343.

1402
A Note on a Method of Computing the Gamma Function by N. L. Gordon and A. H. Flasterstein (RCA); *J. Assoc. Comp. Mach.*, vol. 7, pp. 387–388; October, 1960.

Using the well-known Gamma-function recursion formula $\Gamma(x+1)=x\Gamma(x)$ and Stirling's asymptotic expansion for $\ln \Gamma(x)$, a computationally convenient expression for $\Gamma(x+1)$ is developed. Estimates of error for both single- and double-precision arithmetic are provided.

1403
Design and Operation of Digital Calculating Machinery by H. H. Aiken (Harvard University); *U. S. Govt. Res. Repts.*, vol. 34, p. 610(A); November 18, 1960. PB 150 300 (order from LC mi$7.50, ph$24.30).

The results of computations completed on the Mark I and Mark IV calculators are reported. The coordinate determination of ground points on aerial photographs is discussed. A mathematical description of a nonlinear eigenvalue problem is presented. The solution of nineteen linear second-order differential equations with two-point boundary conditions is reported.

1404
Resultant Procedure and the Mechanization of the Graeffe Process by E. H. Bareiss (Argonne Natl. Lab.); *J. Assoc. Comp. Mach.*, vol. 7, pp. 346–386; October, 1960.

A procedure for finding all the zeros of polynomials with real coefficients, irrespective of the multiplicity of zeros or the presence of cyclotomic polynomials as factors of the original polynomial, is developed. To obtain the absolute values of the zeros, the method relies on a modification of the Graeffe root-squaring algorithm. Operations depending on the theory of resultants are used to separate roots having the same modulus. The numerical stability of the method is established and the question of error is discussed.

1405
On Programming the Numerical Solution of Polynomial Equations by K. W. Ellenberger (North American Aviation); *Commun. Assoc. Comp. Mach.*, vol. 3, pp. 644–647; December, 1960.

The basic Bairstow and Newton-Raphson iteration techniques for computing the roots of polynomials are reviewed. By applying the recommended scaling and inversion rules, a high degree of accuracy and rapid convergence are realized. The method works well for a limited multiplicity of roots. When a root or pair of roots is found, the reduced polynomial from the synthetic division may be used as a new polynomial for subsequent roots. The roots thus obtained expand to a check polynomial agreeing very closely with the original function. Numerical examples are given to illustrate how the methods described circumvent typical pitfalls.

1406
The Pseudoinverse of a Rectangular or Singular Matrix and Its Application to the Solution of Systems of Linear Equations by T. N. E. Greville (Social Security Administration); *SIAM Rev.*, vol. 1, pp. 38–43; January, 1959.

An elementary exposition of the concept of the pseudoinverse of an arbitrary matrix, a generalization of the notion of the inverse of a nonsingular matrix, is presented. The discussion is limited to matrices with real elements, but is easily extended to the complex field. An application to the solution of systems of equations is described.

1407
Dominant Eigenvectors of a Class of Test Matrices by J. S. Lamont and R. A. Willoughby (IBM Res. Ctr.); *SIAM Rev.*, vol. 1, pp. 64–65; January, 1959.

The following problem arises in elementary particle physics: given an irreducible set of three Hermitean matrices $\vec{\sigma}=(\sigma_x, \sigma_y, \sigma_z)$ satisfying cyclic commutation rules, to find the eigenvalues and dominant eigenvectors of the matrix $\vec{k}\cdot\vec{\sigma}$ where $\vec{k}$ is a real vector. The solution for a specific class of test matrices is presented.

1408
Note on Eigenvalue Computation by J. F. Andrus (Lockheed Aircraft Corp.); *Commun. Assoc. Comp. Mach.*, vol. 3, p. 617; November, 1960.

A simple lemma encompassing the fundamental property of the method of unitary triangularization of matrices is stated. By means of the lemma, a procedure is outlined

whereby a matrix may be reduced to triangular form and its determinant may be evaluated by means of a set of unitary transformations.

1409
A Class of Triple-Diagonal Matrices for Test Purposes by P. A. Clement (State College of Washington and NBS); *SIAM Rev.*, vol. 1, pp. 50–52; January, 1959.

A class of triple-diagonal matrices of arbitrary order (continuant or Jacobi matrices), where the main diagonal consists of zeros and the sub- and super-diagonals of nonzero elements, is discussed. Exact inverses are given, together with the eigenvalues for some special cases.

1410
On Preconditioning Matrices by E. E. Osborne (Space Tech. Labs.); *J. Assoc. Comp. Mach.*, vol. 7, pp. 338–345; October, 1960.

Some of the difficulties are discussed which are encountered in obtaining the eigenvalues of a matrix due to the fact that some of the eigenvalues are small compared with the norm. An iterative procedure for reducing the norm by similarity transformations using diagonal matrices is developed. Test cases using a FORTRAN-written program indicate that the convergence of the procedure is rapid.

1411
Reprint of a Note on Rounding-Off Errors by G. E. Forsythe (Stanford University) *SIAM Rev.*, vol. 1, pp. 66–67; January, 1959.

In the integration of smooth functions the round-off errors are generally not distributed like independent random variables. A random rounding-off procedure is proposed which makes the round-off error a true random variable, with a mean value of zero and a mean-square value not greater than $\frac{1}{4}$. The method can be reasonably simulated in machine computation.

The Method of Characteristics for Isoenergetic Supersonic Flows Adapted to High-Speed Digital Computers—*see* 1443.

1412
A Simple Technique for Coding Differential Equations by R. Sefton and R. Vaillancourt (University of Ottawa); *Commun. Assoc. Comp. Mach.*, vol. 3, pp. 616–617; November, 1960.

A technique for solving an arbitrary set of first-order ordinary differential equations is described. The operations of input-output, initialization, and choice of step size are incorporated as a subroutine whose parameters vary from problem to problem. Computing the derivatives from the equations differs for each problem, and this part of the program must be written anew for each problem. By using a language such as FORTRANSIT on the IBM 650, the standard part of the program may be easily dovetailed in as a subroutine.

1413
Solving Integral Equations on a Repetitive Differential Analyzer by R. Tomović and N. Parezanović (Boris Kidrich Inst. of Nuclear Sciences, Belgrade, Yugoslavia); IRE Trans. on Electronic Computers, vol. EC-9, pp. 503–506; December, 1960.

Methods of practical solution of integral equations on electronic differential analyzers are not well developed. In those cases where such methods have been outlined, special and costly additional equipment is required. Results are presented which show that practical solution of integral equations is possible using a repetitive differential analyzer of convenient design.

1414
Finite Transforms by W. P. Reid (Michigan State University); *SIAM Rev.*, vol. 1, pp. 44–46; January, 1959.

A general discussion of finite transforms whose kernels are obtained from a Sturm-Liouville system is given. Comparison is made with the classical separation of variables procedure, and an example of the heat equation with a boundary condition involving a prescribed function of time is solved.

1415
A New Active-Passive Network Simulator for Transient Field Problems by W. J. Karplus (University of California); Proc. IRE, vol. 49, pp. 268–275; January, 1961.

A computer system which represents a hybridization of analog and digital techniques is described. A wide variety of problems characterized by partial differential equations are simulated by means of a network consisting only of positive and negative resistors. No reactors are used, the negative resistors being realized with the aid of dc amplifiers. Unlike conventional analog methods, the time as well as the space variables are approximated by finite-difference expressions; thus the solution proceeds stepwise in time as on a digital computer. Only implicit difference equations are instrumented, so that there is no danger of computational instability no matter how large the time increment. Among the field problems treated by this computer are problems characterized by the diffusion equation, the wave equation, the biharmonic (beam) equation, and various modified forms of these equations.

1416
Error Bounds in Finite-Difference Approximation to Solutions of Symmetric Hyperbolic Systems by H. F. Weinberger (University of Maryland); *J. Soc. Indust. Appl. Math.*, vol. 7, pp. 49–75; March, 1959.

The majority of linear initial-value problems occurring in mathematical physics can be put in the form of symmetric hyperbolic systems. An important property of these systems is their possession of an energy inequality. The given system is replaced by a related finite-difference problem, a solution of the latter constituting an approximate solution to the former. Error bounds are derived utilizing the energy inequalities for the two related systems. In order to obtain explicit bounds, attention is restricted to one particular finite-difference approximation, but similar error bounds can be constructed for other approximating systems.

1417
An Eigenfunction Series Solution of a Certain Hyperbolic Partial Differential Equation by E. J. Scott (University of Illinois); *SIAM Rev.*, vol. 1, pp. 160–166; July, 1959.

The differential equation of the form $(A(x)u_x)_x - B(x)u = C(x)u_{tt}$ with initial conditions $u(x, 0) = u_0(x)$, $u_t(x, 0) = u_1(x)$ and two boundary conditions of the form $k_1u(a, t) + k_2u(b, t) + k_3u_2(a, t) + k_4u_x(b, t) = f(t)$ is considered. The problem is first transformed to one with homogeneous boundary conditions. Application is made to the motion of a fluid enclosed between two pulsating concentric-circular cylinders.

1418
Techniques for Enumerating Veblen-Wedderburn Systems by E. Kleinfeld (The Ohio State University); *J. Assoc. Comp. Mach.*, vol. 7, pp. 330–337; October, 1960.

Veblen-Wedderburn systems are algebraic systems which may be used to coordinate affine planes and thereby projective planes. Basic theorems related to the systems are derived, and the computational procedure for determining the total number of Wedderburn systems and the number of isomorphic classes is outlined.

1419
Linear Recurring Sequences by N. Zierler (M.I.T. Lincoln Lab.); *J. Soc. Indust. Appl. Math.*, vol. 7, pp. 31–48; March, 1959.

After some preliminaries on general recurring sequences on elements of a finite field, a determination of the periods of the members of the family of sequences satisfying a given linear recurrence is given. Characterizations of m sequences (sequences of maximum period generated by linear recurrences) and a discussion of their properties with special reference to their autocorrelation functions are presented.

1420
The Enumeration of Trees by Height and Diameter by J. Riordan (Bell Telephone Labs.); *IBM J. Res. & Dev.*, vol. 4, pp. 473–478; November, 1960.

The enumeration of trees is simplified and elaborated with the help of the concepts of height and diameter. The height of a rooted tree is the length of the longest path from the root, while the diameter of a (free) tree is the length of the longest path joining two endpoints. Enumerations are given for the case in which a fixed number of points are distinctively labelled, thus generalizing the classical concepts of all points alike and all points unlike. The numerical tables furnished extend to trees with 20 points all alike, and to trees with 10 points all unlike.

1421
On Moore Graphs with Diameters 2 and 3 by A. J. Hoffman and R. R. Singleton (General Electric Co.); *IBM J. Res. & Dev.*, vol. 4, pp. 497–504; November, 1960.

The existence of connected, undirected graphs homogenous of degree d and diameter k, having a maximal number of nodes according to a certain definition, is treated. It is shown that for $k=2$, unique graphs exist for $d=2$, 3, 7, and possibly for $d=57$, but for no other degree. For $k=3$, a graph exists only for $d=2$. The characteristic roots and vectors of the adjacency matrix of the graph are utilized in the proofs.

Decoding Nets and the Theory of Graphs—*see* 1354.

9. PROBABILITY, INFORMATION THEORY, AND COMMUNICATION SYSTEMS

1422
A General Program for the Analysis of Surveys by F. Yates and H. R. Simpson (Rothampstead Experimental Station); *Computer J.*, vol. 3, pp. 136–140; October, 1960.

A generalized program for the basic analysis of experimental surveys is described. Data is entered on punched cards which are processed on input, thus saving storage and enabling the number of units in a survey to be virtually unlimited. Variates for each unit are stored temporarily and derived variates constructed where necessary. Mathematical functions are then constructed. Tabulations may be quantitative or qualitative. Special minor subroutines provide various optional features.

Statistical Analysis of Certain Binary Division Algorithms—*see* 1332.

1423
Inductive Proof of the Simplex Method by G. B. Dantzig (RAND Corp.); *IBM J. Res. & Dev.*, vol. 4, pp. 505–506; November, 1960.

A new proof, based on induction, of the Simplex method of solving linear programs is derived. This is in contrast to the classical proof of the existence of an optimal basis obtained from a perturbation of the constant terms. A more elementary proof of the fundamental duality theorem by means of the simplex method is indicated. It is shown that there exists a finite chain of feasible basis changes resulting in either an optimal feasible solution or else in an infinite class of feasible solutions, such that the objective form tends to minus infinity.

1424
An Algorithm for the Assignment Problem by R. Silver (M.I.T. Lincoln Labs.); *Commun. Assoc. Comp. Mach.*, vol. 3, p. 605; November, 1960.

The assignment problem is formulated and briefly discussed. In one application the problem is the assignment of men to jobs for maximum total production. An efficient algorithm for its solution is presented in the ALGOL code. An empirical relation is given between solution time and the size of the problem, based on extensive experiments run on a computer.

1425
Systems of Linear Relations by R. A. Good (University of Maryland); *SIAM Rev.*, vol. 1, pp. 1–31; January, 1959.

A detailed exposition of linear inequalities is given. The main chapter headings are: "Weighted Sums and Weighted Averages," "The Key Theorem and Its Applications," "Game Theory," "Self-Dual Systems," "Linear Programming Theory," "Linear Programming and Game Theory," "Nature of the Set of Solutions of a System of Linear Inequalities."

1426
Integer Programming Formulation of Travelling Salesman Problems by C. E. Miller, R. A. Zemlin (Standard Oil Co. of California), and A. W. Tucker (Princeton University); *J. Assoc. Comp. Mach.*, vol. 7, pp. 326–329; October, 1960.

The formulation of a generalization of the traveling salesman problem, in which the salesman may return to his home city more than once, is discussed in terms of integer programming techniques. The linear form to be optimized and the constraints to be satisfied are detailed. Machine examples using the all-integer algorithm of Gomory are described. In many cases the machine failed to find a solution. The development of more efficient algorithms, it is hoped, will extend the range of usefulness of the model.

1427
Optimizers: Their Structure by R. F. Wheeling (Socony Mobil Oil Co.); *Commun. Assoc. Comp. Mach.*, vol. 3, pp. 632–638; December, 1960.

A general discussion of the problem of optimization is attempted. The central problem of optimization is the choice of a suitable model of the system to be optimized. A procedure of selecting a model and recursively improving it is discussed. This dynamic improvement of the model may also be used to decide whether more data are required from the system, to decide what data are required, and to estimate the optimal input to the system. It is concluded that in the present state of knowledge only heuristic rules are in general available to improve models of systems.

1428
Digital Simulation of Discrete Flow Systems by C. J. Moore and T. S. Lewis (Illinois Inst. Tech.); *Commun. Assoc. Comp. Mach.*, vol. 3, pp. 659–660; December, 1960.

Programming a computer to simulate two discrete-flow systems of randomly-moving items by means of statistical models is described. The two systems are a package-handling plant whose capacities and processing rate are to be optimized, and an air-traffic-flow and -control system in which the effects of alternate control decisions are compared. The emphasis is on the programming methods required for such simulation studies.

1429
Methods for Obtaining Complete Digital Chains by B. Lippel and I. J. Epstein (ASRDL); *U. S. Govt. Res. Repts.*, vol. 34, p. 610(A); November 18, 1960. PB 149 383 (order from LC mi$2.40, ph$3.30).

Two methods for obtaining complete digital chains (chain-coded sequences which contain all the possible permutations of p kinds of symbols taken n at a time) are described and proved. A generalized permutation method for complete chains is also given. Possible engineering applications for chain codes are considered.

1430
Cyclic Codes for Error Detection by W. W. Peterson (University of Florida) and D. T. Brown (IBM Corp.); PROC. IRE, vol. 49, pp. 228–235; January, 1961.

Cyclic codes (chain codes, ring codes, shift register codes) are defined and described from a new viewpoint involving polynomials. The basic properties of Hamming and Fire codes are derived. The potentialities of these codes for error detection and the equipment required for implementing error-detection systems using cyclic codes are described in detail.

1431
Binary Codes for Error Control by W. W. Peterson (University of Florida); *Proc. NEC*, vol. 16, pp. 15–21; 1960.

The model of a communication channel used in the study of binary codes, its relation to real data-transmission channels, and assumptions about noise are reviewed briefly. Then several error-correcting codes are listed with a brief description of their error-correcting capabilities and the feasibility of their implementation. The most promising class of codes for implementation of the present are cyclic codes. The burst-error-correcting cyclic codes are especially attractive from the point of view of implementation and a simplified block diagram for implementing a practical example of such a code is shown and discussed.

1432
A Bound for Error-Correcting Codes by J. H. Griesmer (IBM Corp.); *IBM J. Res. & Dev.*, vol. 4, pp. 532–542; November, 1960.

Two new bounds for the code word length n required to obtain a binary group code of order 2^k with mutual distance d between code words are derived. The results are compared with previously known bounds and shown to be improvements for certain ranges of k and d. Values of k and d for which one of the bounds can be achieved are given; in such cases a condition satisfied by the structure of the resulting codes is presented.

1433
A Survey of Digital Methods for Radar Data Processing by F. H. Krantz and W. D. Murray (Burroughs Corp.); *Proc. Eastern Joint Computer Conf.*, pp. 67–82; December 13–15, 1960.

The growing number of declassified techniques for automatic processing of radar data by digital means is reviewed. Emphasis is placed upon signal time-sampling and quantization, integration methods, rejection of stationary targets, radar trigger manipulation, and a new high-speed device for recording digitized radar video. These techniques are discussed individually and are also shown combined in a hypothetical radar data-processor design.

1434
Digital Simulation in Research on Human Communication by E. E. David, Jr. (Bell Telephone Labs.); PROC. IRE, vol. 49, pp. 319–329; January, 1961.

Digital simulation is discussed as a powerful tool in uncovering the basic properties of new or proposed communications principles, particularly those involving coding of visual or auditory information. Operating on digitalized speech or pictorial signals, a stored program computer can perform processing equivalent to any coding. The output signals so produced can then be made available for subjective evaluation, thereby removing the necessity for premature instrumentation to produce samples for viewing or listening. This technique owes its efficacy to: 1) the availability of computers fast enough to accomplish the proc-

essing in a reasonable time scale, 2) the existence of high quality translators to implement the flow of continuous signals in and out of the computer, and 3) the creation of compiling programs which allow uninitiated investigators almost immediate access to computer facilities, and which keep programming effort low. Simulation is assuming an increasing role in communications research.

1435
High Speed Data Transmission Systems by R. G. Matteson (Stromberg-Carlson Co., Div. of General Dynamics Corp.); *Proc. Eastern Joint Computer Conf.*, pp. 97–109; December 13–15, 1960.

With the rapid increase in the use of digital computers for business, scientific, control, and military applications, requirements are created for the high-speed transmission of data so that these computers may be utilized for more, and diversified, applications and so that computers may be utilized a greater percentage of the time. Areas of application of high-speed data transmission equipment are discussed, and equipment is developed for the transmission of digital information over standard telephone lines at 2400 bits per second is described. One of the units considered is a unique type of magnetic tape transport designed to operate in two modes, a stepping asynchronous mode or a continuous, synchronous mode.

1436
Digital Data Communication Techniques by J. M. Wier (Bell Labs.); PROC. IRE, vol. 49, pp. 196–209; January, 1961.

The current status of digital data-communication techniques is reviewed. The majority of digital communications are now carried on the teletypewriter networks. New modulation techniques, higher transmission speeds, and more effective error-control schemes are being provided in systems now being introduced in the digital communications area. A considerable body of theoretical work has been completed which provides a base for most of the activity in the field; however, application of much of this work, until recently, has often been uneconomical in the current state of the art. This is notably true in the case of error control and the practical achievement of theoretical channel capacities, but the problem also arises in other areas. The work towards a truly automatic central data processing system has been slow, but a number of projects are now showing the way in this area. These include reservation systems, centralized banking, central inventory control, air traffic control, and above all, centralized military communications and control systems.

1437
The Testing of Digital Data Transmission Channels and Circuits by R. G. Enticknap (M.I.T. Lincoln Lab.); *Proc. NEC*, vol. 16, pp. 66–71; 1960.

The background, aims and methods of a program of work on data-transmission research are described. The work, which has been concentrated primarily on telephone circuits of a type generally available, aims at a better understanding of these transmission media, particularly with respect to the additive and multiplicative noise which appear to constitute the major source of errors in current data transmission systems. The program is considered in a number of phases: 1) the measurement of errors incurred by certain data-transmission systems used over specific circuits, 2) the measurement of additive noise on specific circuits, and 3) the measurement of dynamic characteristics of certain circuits. While no results of these measurements are given, they are referenced in other papers. An attempt is made to summarize the program to date and to cover the additions to and extension of the work in the near future.

1438
Error Recording and Analysis for Data Transmission by C. M. Melas and E. Hopner (IBM Corp.); *Proc. NEC*, vol. 16, pp. 54–65; 1960.

Errors recorded to help in the evaluation of error-correction or -detection schemes necessary for reliable telephone transmission are reported. The error-checking and -recording equipment designed for these tests had to be simple enough to be readily transportable, and yet had to record all pertinent data for subsequent compilation by the computer. A repetitive 8-bit combination was used for transmission, and error checking was accomplished without the use of a synchronizing signal at the receiver. The character generator is also able to produce a long pseudo-random sequence, which is duplicated at the receiver for bit-by-bit error detection, providing a closer approximation of actual operating conditions. The data thus acquired is reduced to provide information on the duration and nature of error bursts and interruptions, as well as the length of error-free transmissions.

1439
Measured Error Distributions on the Bell A-1 Facility over Various Media by E. J. Hofman (M.I.T. Lincoln Lab.); *Proc. NEC*, vol. 16, pp. 37–44; 1960.

Error distributions obtained by experiments carried out using an automatic digital data-error recorder are reported. Media include several land line telephone circuits and the Hawaiian cable. Both data- and sync-error rates are shown, and their temporal and amplitude characteristics are examined. It is shown that digital errors are of a burst nature both within a word and over many words. Distributions are given for the number of bits in error in a word error, duration of bursts and the probability of continuous error and error-free periods. Conversion probabilities of a bit from mark to space and space to mark are shown. The effectiveness of parity checking as an error-detecting device is also evaluated.

10. SCIENCE, ENGINEERING, AND MEDICINE

The Printed Motor—*see* 1362.

1440
Some Aspects of Simulator Design by J. M. Dempsey (Panellit Ltd.); *Computer J.*, vol. 3, pp. 158–161; October, 1960.

The operation of the Berkeley Nuclear Power Station Simulator is outlined. The need for simulators to be realistic in appearance to be of value in training is emphasized. A system method for notation and scaling is required in order to provide for the ready solution of fuel temperature control equations. Simulation speed has proven in practice to be faster than real-time operation.

1441
An Analysis of a Hydro-Electric System by P. F. King and D. A. Peel (British Aluminium Co., Ltd.); *Computer J.*, vol. 3, pp. 161–163; October, 1960.

A mathematical model used to simulate the operation of a hydro-electric scheme in Scotland is described. A study of various alternate ways of operating the system and an evaluation of the effect on over-all power production of various proposed modifications is thereby facilitated.

1442
On Iterative Factorization in Network Analysis by Digital Computer by W. H. Kim, C. V. Freiman, W. Mayeda, and D. H. Younger (Columbia University); *Proc. Eastern Joint Computer Conf.*, pp. 241–253; December 13–15, 1960.

The need to determine the sum of all tree-admittance products occurs in almost all applications of topological network theory. A method of obtaining this sum through an iterative factorization of the sum of tree admittance products of successively more complex subnetworks is described. Computational efficiency is achieved in that: 1) it is not necessary to test sets of branches for the presence of circuits; and 2) it is not necessary to calculate each tree-admittance product. A digital computer program has been developed for use on an IBM-704 which accommodates networks of up to 13 nodes and 77 branches. The program is designed to reduce computation time when the present network corresponds to a minor modification of the previously investigated network. Use is made of a lexicographic enumerating function to develop working constants for networks of various size. Estimates of computing time and flow-charts of the program are included, as is a table of the number of unique partitions of an n-element set ($n=1, 2, \cdots, 12$).

DYANA—A Computer Program for the Automatic Analysis of Dynamic Systems—*see* 1372.

A New Active-Passive Network Simulator for Transient Field Problems—*see* 1415.

1443
The Method of Characteristics for Isoenergetic Supersonic Flows Adapted to High-Speed Digital Computers by F. E. Ehlers (Boeing Airplane Co.); *J. Soc. Indust. Appl. Math.*, vol. 7, pp. 85–100; March, 1959.

By a transformation of the independent variables in the characteristic equations of isoenergetic supersonic flows, the coefficients in the differential equations are simplified to rational algebraic functions, thus considerably reducing the calculating time required for each step of the numerical integration process. The equations are expressed in difference form and iteration procedures similar to those developed for the classical equations

are presented. Equations for the interaction of Mach lines with a shock wave, with a free surface, and with a solid boundary are given in terms of the new variables. An application is made to the design of perfect nozzles for a supersonic wind tunnel.

1444
Numerical Weather Prediction by N. A. Phillips (M.I.T.); *Advances in Computers*, Academic Press, Inc., New York, N. Y., vol. 1, pp. 43–90; 1960.

The magnitude of the weather prediction problem is shown to make microscopic forecasts over a large area impossible, even with large-scale computers. The gathering, limitations, and analysis of input data is discussed, and the mathematics of two main forecasting systems, the hydrostatic and the geostrophic, are summarized. Model approximations and allowance for frictional and nonadiabatic effects are described. Computational techniques presented include the establishment of computational stability and the inversion of elliptic operators in the geostrophic system. The accuracy of presently available numerical forecasts is assessed.

1445
A Computer for Weather Data Acquisition by P. Meissner, J. A. Cunningham (NBS) and C. Kettering (U. S. Weather Bureau), *Proc. Eastern Joint Computer Conf.*, pp. 57–66; December 13–15, 1960.

A small specialized computer designed for use as the control component in an automatic weather station is described. The computer is intended as a research tool for exploring the application of automatic data processing equipment to this type of problem. Basically, the computer must sample a number of weather-sensing instruments, suitably process the instrument data, and prepare outputs in the form of local displays and teletype messages. Since the machine is internally programmed, it will afford a wide latitude in the processing of input data including the simultaneous comparison of results obtained in different ways. The design of a computer for this application was felt to be justified on the basis of a number of special requirements. Among these are: 1) parallel inputs from a number of separate sources, 2) multiple outputs in several forms, 3) concurrent operation of input, output, and data processing functions, 4) extensive reference tables with special instructions for their use, 5) limited arithmetic capability, 6) three-digit word length, and 7) computing speed which is not necessarily high.

1446
Simulation of Full-Scale Multi-Stage Batchwise Chemical Plant by P. V. Youle (Imperial Chemical Industries, Ltd.); *Computer J.*, vol. 3, pp. 150–157; October, 1960.

The simulation is described of a hypothetical multi-stage batchwise chemical process, having seventeen vessels in four stages, on a Ferranti Mercury Computer. This simulation forms part of a hierarchy of simulations in which the simulation of a chemical reactor forms part of the simulation of the full-scale program, which in turn forms part of the simulation of the whole corporation's activities. Results from the simulation demonstrate their value for plant management decisions.

1447
The Beam-Programming System of the Bevatron by H. G. Heard (Levinthal Electronic Products); IRE TRANS. ON NUCLEAR SCIENCE, vol. NS-7, pp. 4–13; December, 1960.

The beam-programming equipment developed for the Bevatron is described. The system provides analog-computer-controlled as well as beam-controlled frequency tracking. Equipment which permits arbitrary adjustment and control of the radial position of the circulating beam at any energy is discussed. Beam-intensity control and the automatic reduction of phase errors in the acceleration system are provided. Units which produce long and short beam pulses and equipment which permits highly-multiple operation are also discussed. The integration of these equipments for extended injection is treated.

Nuclear Event Pattern Recognizer—*see* 1385.

11. ANALOG AND HYBRID COMPUTERS

1448
Algebraic Function Calculations Using Potential Analog Pairs by M. L. Morgan (Electro Scientific Industries Inc.); PROC. IRE, vol. 49, pp. 276–282; January, 1961.

A new potential analog method is described in which both the magnitude and the angle of functions of a complex variable are calculated from their pole-zero plots. The method is based on a pair of "factor analog" voltage distributions in a complex plane, on which voltage measurements at the zeros and poles of the function are used for the calculation.

1449
Application of the Magnetoresistance Effect to Analog Multiplication by J. M. Hunt (General Precision, Inc.); *Proc. NEC*, vol. 16, pp. 619–628, 1960.

Certain materials exhibit a pronounced increase in resistivity when placed in a magnetic field; the use of resistance elements constructed of these materials as the control elements in modulators or electronic multipliers for analog computation is discussed. Straightforward embodiment of this principle in a multiplier results in a device of seriously limited accuracy and usefulness. However, a number of relatively simple techniques have been developed which permit the attainment of accuracy considerably in excess of that of contemporary electronic multipliers. The simple and inexpensive magnetoresistance multiplier is most directly competitive with the widely used servo multiplier, but offers the very important advantage of complete freedom from the objectionable nonlinear effects of the servo multiplier which result from backlash, static friction, wire-to-wire resolution, etc. The magnetoresistance multiplier is unusually adaptable to design compromise to permit exchange of one desirable characteristic for another (for example, sacrifice of static accuracy to obtain improved frequency response). Factors underlying the optimum choice of magnetoresistance materials are described and possible methods of extending currently attainable performance of the device are outlined.

1450
Analog Representation of Poisson's Equation in Two Dimensions by R. J. Martin N. A. Masnari, and J. E. Rowe (University of Michigan); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-9, pp. 490–496; December, 1960.

A new analog device, called a Poisson cell, which aids in obtaining solutions to either Laplace's equation or Poisson's equation is described. The cell may be used to simulate such potentials as electric potential, magnetic potential, gravitational potential, and the velocity potential of irrotational flow; it has applications in the fields of hydrodynamics, head conduction, and aerodynamics. The cell is a solid volume-conducting medium made from a homogeneous mixture of hydrostone and graphite. Electrode configurations may be painted on the surface with conducting paint or imbedded directly in the structure. In the case of Poisson's equation, where $\nabla^2\phi(x, y) = f(x, y)$, the function $f(x, y)$ is simulated by injecting currents into the underside of the cell. The application of the Poisson cell to numerous problems and in particular to problems in electron flow is discussed in detail, along with the incorporation of the cell into either an analog computer system or a combined analog-digital computer system.

1451
A New, Solid-State, Nonlinear Analog Component by L. D. Kovach and W. Comley (Douglas Aircraft); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-9, pp. 496–503; December, 1960.

A passive, nonlinear device is described which, when used with operational amplifiers, provides the means for obtaining a large class of functions. These functions are obtained to a degree of accuracy and reliability not previously possible with a simple, economical device. A basic varistor squaring unit is utilized. The unit has been compensated for the various types of error inherent in the varistor itself, and is capable of providing approximately fifteen of the most basic and commonly used nonlinear functions.

1452
A New Technique for Analog Integration and Differentiation by M. A. Thomae (Comisión Nacional de la Energía Atómica, Buenos Aires, Argentina); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-9, pp. 507–509; December, 1960.

A technique which enables an approach to ideal analog integration or differentiation by means of passive elements only is described. A series of RC circuits in cascade, decoupled, provides the first, second, third, etc., integrals or derivatives of the input function. If the outputs of each are fed to an analog summing amplifier, its output becomes arbitrarily close to the ideal integral or derivative of the input function as the

number of RC stages is raised indefinitely. An integrator has been built to check the theory, and results are given.

1453
An Analog RC Integrator with a Digital Output by R. J. Jarrett (General Electric Co.); *Proc. NEC*, vol. 16, pp. 611–618; 1960.

In inertial guidance systems, one of the critical components is the integrator used to integrate the accelerometer output to produce velocity. In addition to the need for increased integration accuracy, it is extremely desirable to obtain an output in a form that can be directly applied to a digital computer. A method is described which will give the accuracy required for inertial guidance systems, in addition to a quantized output. The output of precision force-balance accelerometers is a dc current proportional to acceleration. This current is applied to an RC integrator (high-gain dc amplifier with capacitor feedback). The integrator output, which is normally proportional to velocity, is sensed, and when it exceeds a preset limit of either polarity, a current pulse of precisely-controlled amplitude and duration is triggered. The pulse is applied to the integrator input with the correct polarity to cancel the output. The "resetting" of the integrator will repeat each time the output exceeds the limit and the reset pulse rate is proportional to the applied acceleration. Each reset pulse is equivalent to an acceleration applied for a fixed time and is therefore equivalent to an increment of velocity. Each time a reset pulse occurs, it is registered in a digital computer which accumulates the total velocity. The normally troublesome errors of the RC integrator are much less significant with this method. A further advantage is that the limit on the RC integrator output at which "resetting" takes place does not have to be precise. Accuracy is primarily due to the stability of the area of the "reset" pulses.

1454
A Digital Differential Shaft-Motion Analyzer by R. L. Lyon (IBM General Products Div.); *Proc. NEC*, vol. 16, pp. 829–834; 1960.

A system permitting measurement of the angle between two points in a dynamic shaft system using digital and incremental techniques is described. Low inertia incremental shaft transducers and transistorized logic are used to resolve $2\frac{1}{2}$ minutes of angle at shaft speeds up to 1400 rpm. Additional logic is used to compare the movement between these two transducers and to sum the difference in a reversible binary counter. The output voltage for a suitable recorder is derived from a digital to analog converter. The output deflection on the recorder is then proportional to the instantaneous angle between the two points in the dynamic shaft system. The analyzer is able to make 200,000 comparisons of position per second. Experiments performed to measure the differential angle in shaft systems with speed ratios of other than 1/1, and to measure the instantaneous velocity of a shaft with the signals from the same incremental transducers are described.

An Analog RC Integrator with a Digital Output—*see* 1453.

12. REAL-TIME SYSTEMS AND AUTOMATIC CONTROL; INDUSTRIAL APPLICATIONS

1455
Computers in Automatic Control Systems by J. G. Truxal (Polytech. Inst. Brooklyn); PROC. IRE, vol. 49, pp. 305–312; January, 1961.

The increasing use of electronic computers as active controllers in modern automatic feedback-control systems is surveyed. The trend from the use of simple electric circuits and elementary analog devices toward high-speed digital computers (or computers employing both digital and analog devices) is accelerated by: 1) the rapidly-growing complexity of modern control systems, 2) the increasing importance of self-adaptive control, and 3) the development of optimizing control. Recent applications of computer control include the automatic control of boiler operation in an electric generating station, the adaptive autopilots for piloted aircraft, and the automatic control of load dispatching in electric power distribution.

1456
Transistorized Building Blocks for Data Instrumentation by J. A. Cunningham and R. L. Hill (NBS); *U. S. Govt. Res. Repts.*, vol. 34, p. 611(A); November 18, 1960. PB 161 569 (order from OTS $2.00).

Several modular transistorized digital circuits that have been used in automatizing many data-recording and preliminary-processing tasks encountered in scientific operations are described. These versatile building blocks can be connected together systematically to form digital circuits that accept raw data from experimental equipment and transpose these data into a form suitable for input to a high-speed electronic computer. These blocks can be used where: 1) data are produced in large volumes; 2) data taking is extremely fast, extremely slow, or extremely precise; 3) a need exists to minimize human error, tedium, and eyestrain; or 4) computation is extensive. This report describes 5 major packages and 8 special-purpose packages.

A Computer for Weather Data Acquisition—*see* 1445.

13. GOVERNMENT, MILITARY, AND TRANSPORTATION APPLICATIONS

1457
Digital Computer Equipment for an Advanced Bombing, Navigation and Missile Guidance Subsystem for the B-70 Air Vehicle by T. B. Lewis (IBM Corp.); PROC. IRE, vol. 49, pp. 313–318; January, 1961.

Transistorized high-speed general-purpose digital computing equipment is described, which is designed as the primary airborne real-time calculating element for the advanced bombing, navigation, and missile guidance subsystem intended for use in the B-70 air vehicle. Because of the critical nature of this application, reliability, maintainability, and flexibility requirements received the utmost consideration in every design phase. Maximum reliability was achieved primarily through the use of a powerful, versatile, parallel main computer section supported by a minimal serial, emergency computer. A new approach to fault location enhances computer maintainability and permits in-flight repair, while flexibility is achieved by means of unique high-speed input-output processing equipment.

Human Decision Making as Related to Air Surveillance Systems—*see* 1390.

The Organization and Program of the BMEWS Checkout Data Processor—*see* 1334.

14. BUSINESS APPLICATIONS

1458
General Purpose Programming for Business Applications by C. C. Gotlieb (University of Toronto); *Advances in Computers*, Academic Press, Inc., New York, N. Y., vol. 1, pp. 1–41; 1960.

The desirability of programming systems for business applications of computers is stressed, and various types of systems such as assemblers, interpreters, compilers, and generators are differentiated. The systems requirements, speed of storage, and checking and maintenance abilities which are necessary for typical data-processing problems are are discussed, and definitions and notations that have arisen in business programming are summarized. File organization and maintenance is treated at length, and some typical business programming systems are described. The relative trading-off between hardware and programming is evaluated.

1459
Problems of the Introduction of Large Scale Data Processing into the Royal Army Pay Corps by L. D. Slater (Royal Army Pay Corps); *Computer J.*, vol. 3, pp. 120–123; October, 1960.

Planning for the RAPC installation to handle the pay accounts for 150,000 soldiers is described. The system is required to be flexible to permit manifold expansion during a national emergency. Each record is from 800 to 2000 characters in length, and the file is to be updated daily for about 10 per cent activity. An IBM-7070 computer will handle three daily updating runs and the output will be under the control of an IBM 1401. The system requires 130 subroutines, about 25 being used for any one record. The subroutines are ordered on tape according to their frequency of use.

1460
Market Research Applications on LEO by J. A. Gosden (LEO Computers Ltd.); *Computer J.*, vol. 3, pp. 142–143; October, 1960.

Examples of market research on LEO and a program providing cross tabulation of survey data are described. The data is checked for internal consistency and the cross tabulations, presented as matrices for each commodity, are prepared at a rate of a half-minute per 1000 items per table. Examples of applications are the determination of consumer attitude towards products and the optimization of delivery schedules.

1461
Market Surveys with a Small Computer by R. L. Cook (Elliot Bros. Ltd.); *Computer J.*, vol. 3, pp. 140–142; October, 1960.

A general-purpose program to perform market-analysis surveys on a small computer is described. Results of interviews are punched on cards, after which the program checks the data for internal consistency, compiles frequency tables, and outputs the results in any given format for photographic reproduction. The program is designed so that desired subsets of the interviewers may be ignored.

1462
The First Year's Production on a Computer, and Future Plans by C. B. Warmington (Albert E. Reed and Co., Ltd.); *Computer J.*, vol 3, pp. 124–127; October, 1960.

Operating experience over the past year on a National-Elliot 405 computer for a large and geographically diversified company is described. The main applications are inventory control and payroll accounting. Future plans include cost and sales analysis and management-by-exception techniques.

1463
A Bank Adopts Automatic Data Processing by R. Hindle (Martin's Bank Ltd.); *Computer J.*, vol. 3, pp. 127–130; October, 1960.

Problems arising in the automation of banking procedures, particularly those due to differences among branch office organizations, are described. The feasibility and present state of character recognition and the telegraphic transmission of banking data are discussed. The system described is paper-tape oriented and is based on random input. The computer will prepare branch records and a complete customer file.

1464
A Progress Report on the Introduction of ADP for Recording Contributions Paid under the New Graduated Pensions Scheme by D. W. Polley (British Ministry of Pentions and Natl. Insurance); *Computer J.*, vol. 3, pp. 117–120; October, 1960.

The use of electronic data processing for handling the records of 20,000,000 contributions to the new British Graduated Pensions Scheme is described. Off-line search units will handle the 2000 expected enquiries on a two-day cycle and an EMIDEC computer will update the main files. The files will be the largest in the U.K. and it is thought that this is the largest data processing undertaking ever to have been planned directly for computer operation, without having been performed by conventional means.

SUBJECT INDEX

AUTHOR INDEX

PGEC News

To All PGEC Chapter Officers and Committee Chairmen

The PGEC *News* section of the TRANSACTIONS is open for your announcements and reports of activities. Deadline is the first of the month, two months ahead of the date of issue. Send all items to the *Editor*.

Binders for TRANSACTIONS Available from IRE

IRE headquarters has made available binders for convenient shelf storage of IRE TRANSACTIONS issues. The binders are of Spanish-grain fabrikoid with gold lettering, and hold 4 inches of TRANSACTIONS, or 24 issues, whichever is smaller.

Cost is $3.00 without imprinting, or up to $4.50 with full imprinting of Name, TRANSACTIONS Titles, and Years. For order form and more details, write to L. G. Cumming, Technical Secretary, Institute of Radio Engineers, 1 East 79 Street, New York 21, N. Y.

THE CHAIRMAN'S LETTER

St. Paul, Minn., April 7, 1961—

Administrative Committee

The PGEC Administrative Commitee ("AdCom") managed to meet on March 22, 1961, without interference this time from the unpredictable New York weather. Reports were presented by the chairmen of the various standing committees, and officers and new members were elected. A great deal of discussion took place, particularly on the needs and means for improving and expanding PGEC services to its members. The highlights of the transacted business will be included in this letter.

First, I should like to welcome Jerome Kennedy, Charles Sankey, and Bob Stewart to the AdCom. They have been elected for three-year terms. Kennedy is a delegate-at-large, while Sankey and Stewart are from IRE Regions 8 and 5, respectively. I should like also to welcome Wally Anderson to his new post as PGEC Vice-Chairman. Thumbnail sketches on these four gentlemen are included in a news story elsewhere in this issue.

Three members of the AdCom are retiring after three years of hard work and excellent service. They are Frank Heart, who is retiring as Vice-Chairman, and Administrative Committee members Keith Uncapher and Dan Haagens. Although each has completed a formal term of office, these are dedicated men who are serving, and will continue to serve, in various positions of responsibility. The AdCom extended a vote of thanks to these men in recognition of their valuable service.

New PGEC Chapters

Two new chapters of PGEC are now fully approved and under way. The Orange County (California) Chapter was organized by Robert F. Geiger. Bill Gunning has been elected Chairman, Charlie Hobbs, Vice-Chairman, and R. Davis, Secretary-Treasurer. The Syracuse (New York) Chapter was organized by J. F. Forney. At this writing, officers are yet to be elected. We wish to congratulate the organizers and officers of these two Chapters, and to convey the best wishes of PGEC for a successful program of Chapter activities.

On the subject of Chapter activities, it is an objective of the AdCom to assist Chapters in preparing interesting and informative programs. It is planned to disseminate information on speakers who are available either nationally or regionally with well-prepared talks for Chapter meetings.

Proposed Divisionalization

The AdCom feels that it is now timely to consider the merits of establishing a suitable divisional organization within PGEC. With approximately 9,300 members, we are the largest professional group in the IRE, and still growing. By setting up divisions or sub-groups serving several suitably defined areas of technical interest, it is felt that PGEC would be better situated to organize special symposia and conference sessions, and to improve the procurement and review of technical papers, both for presentation and for publication.

The divisionalization concept has the endorsement of the IRE Professional Groups Committee, and the larger professional groups are encouraged to proceed in this direction. This is considered a sounder approach than to allow the proliferation of countless new professional groups, of which there are already 28 within the IRE. Divisionalization has been instituted in the group on Electron Devices. Other professional groups are studying the idea. Accordingly, the AdCom has advised me to form an ad hoc committee to study this subject and to report the initial findings in August. The goal is to organize in such a way so as to keep PGEC dynamic and responsive to the changing needs of the electronic computer field.

Anticipating the need for divisionalization, TRANSACTIONS Editor Howard Tompkins has proposed to the AdCom that several associate editorships be established. The AdCom has approved the appointment of Dr. John E. Sherman as Associate Editor for Analog and Hybrid Computation. One or two additional appointments will be announced in the near future.

Scope of PGEC

The technical areas falling within the scope of PGEC have been reviewed by the AdCom, and agreement was reached at the March 22 meeting on a revised statement of scope. This new statement is as follows:

The scope of the IRE Professional Group on Electronic Computers and its publications encompasses:

(a) all aspects of design, theory, and practice relating to systems for digital and analog computation and information processing;
(b) components and circuits for digital and analog systems, including techniques for accomplishing the functions of logic, arithmetic, storage, control, mass data storage, input, output, and external communication in such systems;
(c) relevant portions of supporting disciplines, including switching theory, symbolic logic, numerical methods, codes and number representation systems, abstract machine or automation theory, bio-sciences, machine learning, pattern recognition, and other extensions of logical machine capabilities;
(d) production, testing, operation, and reliability of digital and analog systems; and
(e) those aspects of application, use, and programming of digital and analog computing devices and information processing systems that relate to their design and operation.

AFIPS Constitution

Full agreement has been reached on the constitution of the proposed American Federation of Information Processing Societies (AFIPS) by the founding societies, IRE, ACM, and AIEE. The final draft is now being assembled by the NJCC Secretary, and it is expected that the constitution of the new Federation can be printed in the next issue of the TRANSACTIONS. According to plans, AFIPS will be formally announced at the WJCC in May, and will officially come into being on July 1 of this year.

PGEC Constitution and Bylaws

As reported last time, certain desirable changes to the PGEC constitution and bylaws have been in process under the guidance of the Constitution and Bylaws Committee, of which Charles Rosenthal is Chairman. The proposed bylaw changes have now been fully approved by the PGEC AdCom and the IRE Executive Committee. The constitutional amendments, however, are still under review by the IRE Executive Committee. It is expected that the constitutional changes together with the bylaws, can be printed in a forthcoming issue of the TRANSACTIONS, for the information of

PGEC members and for their possible criticism.

IRE Computers Committee

I am pleased to report that Charlie Hobbs has been appointed Chairman of the very important IRE Computers Committee (Technical Committee 8). This is the IRE committee responsible for technical standards activities in the computer area. It works closely with corresponding groups in other societies to arrive at mutually agreeable standards, specifications and definitions. The various IRE technical committees are responsible to the IRE Standards Committee rather than to any of the professional groups. However, the professional groups are strongly encouraged by the IRE to support the technical committees and to assist in staffing them with qualified, dependable people. We congratulate Charlie on this appointment, for which he is extremely well qualified by his background and experience, and pledge the full support and cooperation of PGEC to the Computers Committee.

Conferences

Dr. Charles F. Spitzer has been serving as PGEC representative on the Technical Program Committee for the 1961 WESCON, with the support of the PGEC Conferences and Symposia Committee, of which Dr. E. C. Johnson is our new Chairman.

PGEC will be a co-sponsor this year of the Non-Linear Magnetics Conference to be held on the West Coast in November. Richard R. Booth will represent PGEC on the Technical Program Committee. More on this later.

PGEC Needs You

In order to carry out an effective program of services to its members, PGEC welcomes volunteers for active participation. Openings exist on the various committees, and in editorial and paper reviewing work, for properly qualified members who are willing to devote time and effort to the cause.

If you would like to volunteer your talents, please write to me. It will be helpful if you can enclose a brief resume indicating something about your education and professional experience. What do you regard as your specialty within the computer field? Please include any specific professional society experience, such as IRE committee work, and offices held. Are there areas of PGEC activity in which you would especially like to participate?

May I hear from you?

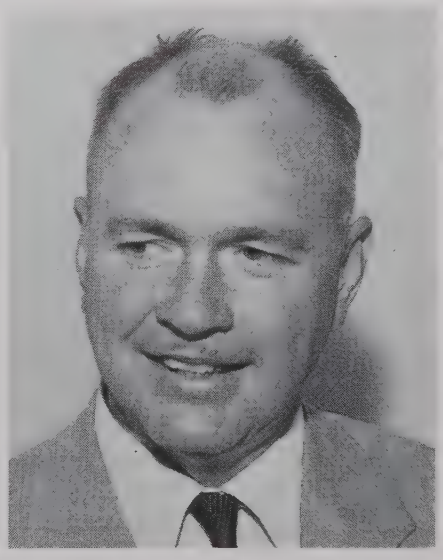
Arnold A. Cohen
Chairman, PGEC

GUEST EDITOR OF FORTHCOMING ISSUE

John McLeod, Guest Editor of the forthcoming special issue of IRETEC on Analog and Hybrid Computers, was born in Hattiesburg, Miss., on February 27, 1911. He received his undergraduate education in New Orleans, La., being graduated from Tulane University in 1933 with the B.S. degree in electrical and mechanical engineering. Subsequent technical courses in instrumentation and automatic control were taken at the University of Chicago, Harvard University, Massachusetts Institute of Technology, and the University of California.

His professional experience has been as Applications Engineer with Taylor Instrument Company; Research and Development Engineer with Leeds and Northrup Co., Philadelphia, Pa. (he holds basic patents on their pneumatic computing controller); General Engineer in charge of the development and operation of the U. S. Naval Air Missile Test Center Analog Computation and Simulation Laboratory; and his present assignment as design specialist in the Flight Performance and Guidance Analysis Group at Convair-Astronautics in San Diego, Calif.

In 1952, Mr. McLeod organized, and was elected chairman of the Simulation Council, a society devoted to improving simulation techniques among users of electronic computers. There are now six regional Councils throughout the United States which are chartered by Simulation Councils Inc., of which Mr. McLeod is currently Secretary. He is also Editor of the *Simulation Council Newsletter*, which appears monthly as a feature of the magazine *Instruments and Control Systems*.



J. McLeod

He is author of the section entitled "Electronic analog computer techniques for the design of servo systems" in the McGraw-Hill "Computer Handbook," currently in press; and has presented numerous technical papers, mostly instrumentation, automatic control, and computers.

Mr. McLeod is currently a member of the Computing Devices Committee of the American Institute of Electrical Engineers, and of the AIEE Committee on Electrical Techniques in Medicine and Biology. He is a member, too, of the American Association for the Advancement of Science. He was Associate Program Chairman for the 1961 Western Joint Computer Conference, and an organizer of the 1961 San Diego Bio-Medical Engineering Symposium. He is a Registered Engineer in the State of California.

NEW PGEC ADMINISTRATIVE OFFICERS AND COMMITTEE MEMBERS

At the annual meeting of the PGEC Administrative Committee, held at IRE Headquarters in New York on March 22, 1961, Dr. Arnold A. Cohen was re-elected Chairman of PGEC, and Walter L. Anderson was elected Vice-Chairman, for the year 1961–62. Charles Sankey, Jerome B. Kennedy, and Dr. Robert M. Stewart, Jr., were elected to the Administrative Committee for three-year terms.

Mr. Anderson, new PGEC Vice-Chairman, is Vice-President and Director of Electronics at General Kinetics, Inc., Arlington, Va. He was born in St. Paul, Minn., in 1922, and holds the degrees of B.E.E. and M.S.E.E. from the University of Minnesota, Minneapolis. He has been in the electronic computer field since 1946, when he joined Engineering Research Associates, Inc., which later became a part of Remington Rand Univac. He has been in his present position since 1955. He is a past Chairman of the Washington, D. C. Chapter of PGEC, and has been on the PGEC Administrative Committee since 1958. He was the PGEC representative on the Technical Program Committee for the 1960 IRE Convention, and has been Chairman of the PGEC Conferences and Symposia Committee.

Mr. Sankey is in the Operational Research Branch of the Canadian National Railways, Montreal. He was born in Nairobi, Kenya, in 1931, and holds a degree in Physics from Oxford University, Oxford, England. He has been an electronics engineer with the Canadian Marconi Company, Montreal, and was head of the Science and Engineering Branch of Adalia Computations, Ltd., prior to his present position.

Mr. Kennedy is Manager of Engineering Sales at Electronic Associates, Inc., Long Branch, N. J. He was born in 1929 and holds the degree of B.S.E.E. from the University of Illinois, Urbana. He was with McDonnell Aircraft, and with the Army Signal Corps, before joining his present company in 1953. He has been active in IRE work and in Simulation Councils, Inc.

Dr. Stewart is Professor of Physics and Electrical Engineering at Iowa State University, at Ames. He is a native of Utah, and holds the degrees of B.S.E.E. and Ph.D. (Physics) from Iowa State University. His extensive teaching and research experience covers a variety of specialities, including digital systems, analog computers and simulators, magnetic film logic, and responsibility for the construction of the ISU digital computer "Cyclone." He has published numerous papers. He is a past chairman of the Des Moines-Ames Section of IRE, and was on the Papers Committee for the 1954 National Electronics Conference.

GLOBAL COMPUTER CONGRESS PLANS SET BY INTERNATIONAL FEDERATION

An international computer conference, to be known as IFIP Congress '62, will be held in Munich, Germany, from August 27 to September 1, 1962, under the sponsorship of the International Federation of Information Processing Societies (IFIPS). The Congress offers computer scientists and other specialists in the information-processing sciences an opportunity for international exchange of ideas and technical knowledge. Plans for the Congress were ratified by the Council of IFIPS at a meeting in Darmstadt, Germany, in February.

Isaac L. Auerbach, representing the National Joint Computer Committee of the United States, and President of the International Federation of Information Processing Societies, states that the IFIP Congress '62 will continue the activities of the International Conference on Information Processing sponsored by UNESCO in Paris, in June, 1959. A global exhibition of computers and other information-processing equipment will be held in the Exhibition Park Theresienhoehe, Munich, at the same time as the Congress.

The following list of officers and chairman have been appointed for the forthcoming Congress:

Professor Dr. Alwin Walther, of the Technische Hochschule in Darmstadt, and Dr. Hans J. Piloty, of the Technische Hochschule in Munich, were designated the Congress '62 General Chairman and Vice General Chairman, respectively. They represent DARA (Deutsche Arbeitsgemeinschaft für Rechen-Anlagen), the German national computer technical society, host of the Congress.

Mr. Auerbach also appointed Professor Dr. R. Sauer of the Technische Hochschule in Munich as Chairman of the Arrangements Committee, and Mr. Albrecht Günter of Siemens-Schuckert, Munich, as Vice Chairman. This committee will be responsible for the organization of all aspects of the Congress in the city of Munich, except for the exhibits. Dr. Heinz Billing, of the Max-Planck-Institut für Physik und Astrophysik in Munich, was appointed Chairman of the Exhibition Committee, with the responsibility of organizing the exhibition and the plant tours.

Professor Dr. Sauer has designated Professor Dr. Hans Thoerner, of the Technische Hochschule in Munich, the President of the Electrical Manufacturers' Association of Germany, as local Chairman for Public Relations for IFIP Congress '62.

Dr. Maurice V. Wilkes, of Cambridge University, England, was appointed Chairman of the Publications Committee. He has designated Miss C. M. Popplewell as Editor of the Congress Proceedings.

Dr. Niels Ivar Bech, of Regnecentralen in Copenhagen, Denmark, was appointed by Mr. Auerbach as Chairman of the Congress Program Committee. His committee consists of the representatives to the IFIPS Council from the member technical societies, except for Dr. E. L. Harder, of the United States, who will serve in place of Mr. Auerbach, and Professor Dr. F. L. Bauer of Germany, who will serve in place of Professor Dr. Walther. Both Mr. Auerbach and Dr. Walther are ex-officio members of this committee. The full list of program representatives follows:

Country and Representative	*Address*
Argentina H. R. Ciancaglini	Facultad de Ingenieria Universite de Buenos Aires Buenos Aires
Australia Dr. John M. Bennett	The Adolph Basser Computing Lab. University of Sydney Sydney
Austria Dr. Heinz Zemanek	Institut fur Niederfrequenztechnik Technische Hochschule Wien
Belgium Prof. M. Linsman	Centre de calcul de l'Universite de Liége 6, Quai Banning Liége
Canada Prof. C. C. Gotlieb	University of Toronto Computation Centre Toronto 5, Ontario
Czechoslovakia Ing. Jiri Kryze	Institute of Information Theory & Automation Czechoslovak Academy of Science Ceskonalinska 25 Prague 6
Denmark Dr. N. I. Bech	Regnecentralen Gl. Carlsbergvej 2 Copenhagen-Valby
England Prof. M. V. Wilkes	University Mathematical Laboratory Corn Exchange Street Cambridge
Finland Prof. Pentti Laasonen	Finland Institute of Technology Helsinki
France J. Carteron	Institut d'Astrophysique 98 Bis. Boulevard Arago Paris 14e
Germany Prof. Dr. F. L. Bauer	Institut für Angewandte Mathematik Universität, Mainz, Jakob Zelder Weg 7 Mainz
Italy A. Ghizzetti	Instituto Nazionale per le Applicazione del Calcolo 7, Piazzale delle Scienze Rome
Japan Prof. H. Yamashita	Japanese Electronic Industry Development Association Nishikubo Tomeocho Minatoku Tokyo
Mexico Dr. Sergio F. Beltran	Mexican Association for Information Processing Centro Electronico de Calculo Ciudad Universitaria Mexico City
Netherlands Prof. Dr. Ir. A. van Wijngaarden	University of Amsterdam Mathematisch Centrum Oudemanhuispoort 4 Amsterdam
Poland Dr. Leon Lukaszewicz	Zaklad Aparatow Mathematycznych Polskie j Akademii Nauk Ul. Koszykowa 79 Warszawa
Spain Prof. J. G. Santesmases	Instituto de Electricidad y Automatica Facultad de Ciencias Ciudad Universitaria Madrid 3
Sweden Mr. Borje Langefors	SAAB Linkoping
Switzerland Dr. A. P. Speiser	IBM Research Laboratory 108 Zurichstrasse Adliswil-Zurich
U.S.A. Dr. E. L. Harder	Westinghouse Electric Corp. Industry Engineering 11 L East Pittsburgh, Pa.
U.S.S.R. Prof. A. A. Dorodnicyn	Computing Centre Academy of Sciences of the USSR I-Academichesky Proezd 28 Moscow B-312

(See the Notices pages of this issue for other details of IFIP Congress '62.)

NEW ASSOCIATE EDITOR

John E. Sherman (M'58), recently appointed Associate Editor for Analog and Hybrid Computers of IRETEC, was born in Brooklyn, N. Y., on January 18, 1922. He received the A.B. degree in mathematics and physics from Hofstra College, Hempstead, N. Y., in 1950. During 1950 and 1951, he attended the University of Oklahoma, Norman, where he did graduate work in physics.

J. E. Sherman

From 1951 to 1952, he was employed by the U. S. Naval Air Missile Test Center, Point Mugu, Calif., where he operated an analog computer. In 1953 and 1954, he was in charge of the analog computer installation at McDonnell Aircraft Corporation, St. Louis, Mo. Since 1954, he has been in charge of all the analog computer installations in the Lockheed Missiles and Space Division, Sunnyvale, Calif.

During 1956 and 1957, Mr. Sherman was the Chairman of the Western Simulation Council. He was a Director of Simulation Councils, Inc., in 1959 and 1960, and the Vice Chairman of the Technical Sessions Committee of the 1960 Western Joint Computer Conference. He is a member of the Association Internationale pour le Calcul Analogique and Chairman of the Board of Directors of Simulation Councils, Inc.

IFIPS COMMITTEE SET UP FOR TERM AND SYMBOL STANDARDS

The Council of IFIPS has authorized the formation of a Committee for the Standardization of Terminology and Symbols; and Isaac L. Auerbach, President of IFIPS and representative of the National Joint Computer Committee of the United States, has appointed G. C. Tootill as Chairman of the new Standards Committee. Mr. Tootill is associated with the Royal Aircraft Establishment in England. All 17 member nations of IFIPS will be represented on his committee.

In other business conducted by the IFIPS Council, President Auerbach appointed Professor Dr. A. van Wijngaarden as Chairman of the Finance Committee for IFIPS. Professor Wijngaarden, the Netherlands delegate representing Nederlands Rekenmachine Genootschap, is associated with the Mathematisch Centrum in Amsterdam.

Professor M. Linsman, of Belgium, has been appointed Editor of the IFIPS Bulletin. He is associated with the Centre Interdisciplinaire de Calcul of the University of Liege.

B. Langefors was appointed Chairman of the IFIPS Admission Committee. Mr. Langefors is associated with the Matematikmaskinnamnden in Stockholm, Sweden.

NEW PGEC CHAPTER

A chapter of PGEC has been established in Orange County, south of Los Angeles, Calif. The first meetings of the new chapter were held on April 13, May 4, and June 1. A program of meetings is now being planned for next fall and winter.

Officers of the new chapter are: William Gunning of Epsco-West, Chairman; Charles Hobbs of Aeronutronics, Vice-Chairman; and Raymond Davis of North American Aviation, Secretary-Treasurer. Committee appointments include Henry Herold of Hughes as Membership Chairman and LeRoy Barter of Beckman Instruments as Arrangements Chairman. Vice Chairman Hobbs will serve as Program and Publicity Chairman.

CALL FOR PAPERS

IRETEC SPECIAL ISSUE ON ANALOG & HYBRID COMPUTERS

The February, 1962, issue of IRE Transactions on Electronic Computers will be a special issue devoted to analog and hybrid analog-digital computers, techniques, and applications. Those who are in a position to contribute an article describing recent progress that has not been widely or completely reported elsewhere are invited to send a full-length preliminary draft, or finished paper, to the Guest Editor for consideration. Titles, authors, and abstracts should be sent as soon as possible. Final selection of papers will be based on full-length manuscripts received no later than July 15, 1961. Authors will be notified by September 1, 1961, and must have their papers corrected and returned to the Guest Editor in final form by October 1, 1961.

John McLeod
Guest Editor, IRETEC
8484 La Jolla Shores Drive
La Jolla, Calif.

Notices

This *Notices* Section is open to all who have an announcement of a conference, symposium, session, publication, or other artifact of interest to the PGEC membership. Please send announcements to the *Editor*, who will put them in the first available issue. The right is reserved to edit the announcements, and to decide whether they indeed are aimed at our audience.

SPECIAL SUMMER PROGRAMS

Several universities are offering special summer programs in areas related to computers. Many of these are in June, and already under way at the time you receive this issue. The later programs that we have knowledge of are as follows:

1) July 10–21 (2 weeks)
Analysis of Nonlinear Systems, directed by Prof. Harold F. Klock
Case Institute of Technology
Write: Herbert B. Schultz, Jr.
Manager of Special Programs
Case Institute of Technology
University Circle
Cleveland 6, Ohio

2) July 10–28 (3 weeks)
Process Control Theory, directed by Prof. Donald P. Eckman,
Case Institute of Technology
[See 1) above]

3) July 17–28 (2 weeks)
Harmonic Analysis for Engineers and Scientists, directed by Prof. Norbert Wiener
University of California at Los Angeles
Write: Department K,
University Extension,
University of California,
Los Angeles 24, Calif.

4) July 31–August 11 (2 weeks)
Digital Control Systems Engineering, directed by Prof. Harry W. Mergler,
Case Institute of Technology
[See 1) above]

5) August 14–25 (2 weeks)
Superconductivity and its Engineering Applications,
Massachusetts Institute of Technology
Write: Prof. Peter Elias, Head,
Dept. of Electrical Engineering,
Room 4-202, M.I.T.,
Cambridge 39, Mass.

COMING MEETINGS PAPERS DEADLINE PAST

EIGHTH DENVER RESEARCH INSTITUTE SYMPOSIUM

The Eighth Annual Symposium on Computers and Data Processing sponsored by Denver Research Institute, of the University of Denver, University Park, Denver 10, Colo., will be held on June 22–23, 1961, at the Elkhorn Lodge in Estes Park, Colo. Please note that the dates are more than a month earlier than in previous years.

The following session topics have been selected:

I. Components
II. Logic Design
III. Philosophy of Computer Design
IV. Computers and Education.

For further information write to
W. H. Eichelberger,
Chairman, Arrangements
Denver Research Institute
University Park.
Denver 10, Colo.

NORTHWEST CONFERENCE

The 1961 Northwest Computing Association Annual Conference will be held on July 21–22, 1961, in Vancouver, British Columbia, Canada. The University of British Columbia is the co-sponsor, and conference sessions will meet on the campus.

For information write to
Conference Information
Northwest Computing Association
Box 836
Seahurst, Wash.

INTERNATIONAL CONFERENCE ON ANALOG COMPUTATION

The Third International Conference on Analog Computation will take place in Belgrade on September 4–9, 1961. This Conference, organized by the International Association for Analog Computation and the Yugoslav National Committee for ETAN, will be divided into four sessions:

1) Theoretical considerations
2) Analog computing equipment
3) Application of analog methods and devices
4) Connection between analog and digital techniques.

The first section will deal with general and specific theoretical problems concerning the principles of analog computation, the characteristics of computing equipment, and the solution of various problems by analog methods. The second section is devoted to practical achievements and experiences in the design and realization of various analog computers and computing elements. The third section will deal with the application of analog computing devices for simulation, computation and analysis in industry, science and engineering. The fourth section will consider the relation between analog and digital techniques, their common aspects and interferences.

Apart from the scientific program, a special entertainment program for the participants of the Conference and their families (visits, excursions, banquets) will also be arranged. An exhibition of analog computing equipment and components will be organized during the Conference in conjunction with the International Fair of Technical Achievements which is held every Autumn in Belgrade. Companies wishing to participate in the exhibition of analog computing equipment are requested to contact the Committee.

The Conference may be attended by all persons interested either as individuals or as elected representatives of scientific institutions or companies. Each person taking part in the Conference is entitled to read a paper which must deal with questions concerning analog computation or related fields.

All correspondence relating to the Third International Conference on Analog Computation should be addressed to
Yugoslav Committee for ETAN
Terazije 23/VII
Belgrade, Yugoslavia

ACM 16th NATIONAL CONFERENCE

The Sixteenth National Conference of the Association for Computing Machinery will be held at the Statler Hotel, Los Angeles, Calif., on September 5–8, 1961. Contributed papers concerned with all phases of analog and digital computation, business applications, and data processing will be presented.

The program will also include a substantial number of invited papers, survey talks, round-table discussions, and a "Hall of Discussions."

Preprints of contributed papers will be available at the time of the meeting.

For further information, write to
Ben F. Handy, Jr.
Litton Systems, Inc.
P. O. Box 1437
Santa Monica, Calif.

INTERNATIONAL CONFERENCE ON MACHINE TRANSLATION OF LANGUAGES AND APPLIED LANGUAGE ANALYSIS

The Conference will be held on September 5–8, 1961 at the National Physical Laboratory in Teddington, England, under the sponsorship of the Autonomics Division.

Papers will be presented by workers engaged directly in research into the machine translation of natural languages and also by those who are concerned with the syntactic or semantic analyses of languages, where such analysis may be of help in achieving machine translation.

For further details write to
Autonomics Division
National Physical Laboratory
Teddington, Middlesex
England

AIEE FALL MEETING

The AIEE will again sponsor a Symposium on Switching Circuit Theory and Logical Design at its Fall General Meeting. This year the meeting is to be in Detroit, Mich., during the week of October 15–20.

Preprints of the papers will be available.

For further information write
Dr. Robert S. Ledley
National Biomedical Research Foundation
8600 Sixteenth Street
Silver Spring, Md.

COMING MEETINGS PAPERS DEADLINES AHEAD

Conference on Nonlinear Magnetics

The 1961 Special Technical Conference on Nonlinear Magnetics is scheduled for November 6–8 at the Statler Hilton Hotel in Los Angeles. This meeting is being sponsored by the IRE Professional Group on Electronic Computers, the Professional Group on Industrial Electronics, and the AIEE. Technical papers in the general field of nonlinear magnetics are solicited. Closing date for papers is August 1, 1961. For further details, contact Dr. Ted Bernstein at Space Technology Laboratories, P.O. Box 95001, Los Angeles 45, Calif.

1961 EJCC CALL FOR PAPERS

The 1961 Eastern Joint Computer Conference, to be held December 12–14 at the Sheraton-Park Hotel in Washington, D. C., is soliciting papers on the theme, "Computers—Key to Total Systems Control."

This theme is to be treated in its broad sense; papers will be welcomed on all advances in computer hardware and concepts leading toward present and future control of industrial, government, defense, and business management systems.

Each person wishing to present a paper at the conference should submit two copies of both a 100-word abstract and a two-page summary to the Program Committee Chairman:

Bruce G. Oldfield
IBM Federal Systems Division
326 E. Montgomery Avenue
Rockville, Maryland

The deadline for submission of abstracts and summaries is June 20, 1961. All submissions should be identified by title, author and affiliation. The full text of papers chosen for presentation must be submitted by September 1, 1961 in order to allow time for publication prior to the conference.

AIEE WINTER MEETING

At the AIEE Winter General Meeting in New York City, January 28–February 2, 1962, the AIEE Computer Systems Subcommittee will sponsor sessions on Kilomegacycle (Gigacycle) Computing Systems (defined simply as systems operating at clock frequencies near and above 1000 megacycles).

Authors are invited to submit papers for these sessions, as well as for other computer sessions at the same meeting, on the following schedule:

July 1, 1961:
Deadline for 100-word abstract and 500-word informal summary. (Send the entire paper if available.)

October 30, 1961:
Deadline for full text of TRANSACTION PAPERS prepared in accordance with the AIEE authors' guide, one copy to be sent to the papers chairman, and four to Mr. E. C. Day, Assistant Secretary for Technical Papers, American Institute of Electrical Engineers, 33 West 39th Street, New York 18, N. Y.

November 25, 1961:
Up to this date papers can be accepted as "Conference Papers" for preprint, but cannot be immediately processed for Transactions. Copies are to be sent to Mr Day and the papers chairman as above.

Correspondence regarding these sessions should be addressed to:

J. H. Wright, Papers Chairman
Division 12
National Bureau of Standards
Washington 25, D. C.

IFIPS CONGRESS '62

The International Federation of Information Processing Societies (IFIPS) will hold a Congress in Munich, Germany, from August 27 to September 1, 1962. The Congress will cover all aspects of Information Processing and Digital Computers including the following:

1) *Business Information Processing*
Data processing in commerce, industry, and administration.
2) *Scientific Information Processing*
Numerical analysis; calculations in applied mathematics, statistics, and engineering; data reduction; problems in operations research.
3) *Real Time Information Processing*
Reservation systems; computer control; traffic control; analog-digital conversion.
4) *Storage and Retrieval of Information*
Memory devices; library catalogs.
5) *Language Translation and Linguistic Analysis*
6) *Digital Communication*
Encoding; decoding; error-detecting and error-correcting codes for digital data transmission.
7) *Artificial Perception and Intelligence*
Pattern recognition; biological models; machine learning, automata theory.
8) *Advanced Computer Techniques*
Logical design; logical elements, storage devices; ultra-high-speed computers; program techniques; ALGOL.
9) *Education*
Selection and training of computer specialists; training of nonspecialists in the use of computers; information processing as a university subject.
10) *Miscellaneous Subjects*
Growth of the information-processing field.

In each category, it is planned to cover, where appropriate, the applications of digital computers, programming, systems design, logical design, equipment, and components.

U. S. authors wishing to offer papers are invited to send abstracts of 500–1000 words to:

Dr. E. L. Harder
Westinghouse Electric Corporation
East Pittsburgh, Pa.,

by September 15, 1961. These abstracts will be considered by the international program committee of IFIPS, and authors of selected abstracts will be invited to submit their complete papers (in French or English) for consideration by the program committee in March, 1962. Authors in other countries should send their manuscripts to the national representative for their country. (See the news story in this issue on page 339 for the full list of program representatives.)

In addition to contributed papers, there will be invited papers, symposia, and panel discussions.

INFORMATION FOR AUTHORS

IRE TRANSACTIONS ON ELECTRONIC COMPUTERS is published quarterly, in March, June, September, and December, with a distribution of over 9000 copies, largely to engineers, logicians, and supervisors in the computer field. Its scope includes the design, theory, and practice of electronic computers and data-processing machines, digital and analog, and parts of certain related disciplines such as switching theory and pulse circuits.

If a paper of widespread interest beyond the computer field is submitted, it will be recommended to the Editor of PROCEEDINGS OF THE IRE for publication. If our reviewers feel that a paper should be submitted to a different IRE TRANSACTIONS, we will so recommend to the author.

Publication time in IRE TRANSACTIONS ON ELECTRONIC COMPUTERS, from receipt of the original manuscript to mailing of the issue, is normally in excess of 5 months, but can be made as little as 3½ months if the occasion demands and the manuscript is carefully prepared.

To avoid delay, please be guided by the following suggestions:

A. Process for Submission of a Technical Paper

1) Send to the appropriate Editor three copies of your manuscript, each copy complete with illustrations. (For Letters to the Editor, two copies will do.)

2) Enclose originals for the illustrations, in the style described below. Alternatively, be ready to send the originals immediately upon acceptance of the paper.

3) Enclose a separate sheet giving your preferred address for correspondence and return of proofs.

4) Enclose a technical biography and photograph of each author, or be ready to supply these upon acceptance of the paper. For biography style, see any IRE journal.

5) If the manuscript has been presented, published, or submitted for publication elsewhere, please so inform the Editor. Our primary objective is to publish technical material not available elsewhere, but on occasion we publish papers of unusual merit that have appeared or will appear before other audiences.

B. Style for Manuscript

1) Typewrite, double or 1½ space; use one side of sheet only. (Good office-duplicated copies are acceptable.)

2) Provide an informative 100- to 250-word summary (abstract) at the head of the manuscript. It will appear with the paper and also separately in PROCEEDINGS OF THE IRE.

3) Provide a separate double-spaced sheet listing all footnotes, beginning with "*Received by the PGEC _____," and "†(Affiliation of author)," and continuing with numbered references. Acknowledgment of financial support is often placed at the end of the asterisk footnote.

4) References may appear as numbered footnotes, or in a separate bibliography at the end of the paper, with items referred to by numerals in square brackets, *e.g.*, [12]. In either case, references should be complete, and in IRE style.

Style for papers: Author (with initials first), title, journal title, volume number, inclusive page numbers; month, year.

Style for books: Author, title, publisher, location, year; page or chapter numbers (if desired).

See this or previous issues for further examples.

5) Provide a separate sheet listing all figure captions, in proper style for the typesetter, *e.g.*: "Fig. 1—Example of a disjoint and distraught manifold."

C. Style for Illustrations

1) Originals for illustrations should be sharp, noise-free, and of good contrast. We regret that we cannot provide drafting or art service.

2) Line drawings should be in India ink on drafting cloth, paper, or board. Use 8½×11 inch size sheets if possible, to simplify handling of the manuscript.

3) On graphs, show only the coordinate axes, or at most the major grid lines, to avoid a dense, hard-to-read result.

4) All lettering should be large enough to permit legible reduction of the figure to column width, perhaps as much as 4:1.

5) Photographs should be glossy prints, of good contrast and gradation, and any reasonable size.

6) Number each original on the back, or at the bottom of the front.

7) Note item B-5 above. Captions lettered on figures will be blocked out in reproduction, in favor of typeset captions.

Mail analog and hybrid computer manuscripts to:
John E. Sherman
Associate Editor, IRETEC
Lockheed MSD
Sunnyvale, Calif.

Mail all other manuscripts to:
Prof. Norman R. Scott
Editor-in-Chief, IRETEC
Dept. of Electrical Engineering
University of Michigan
Ann Arbor, Mich.

(*Continued from outside cover*)

Affiliate Status

Members of the professional societies listed below, who are not IRE members, may become AFFILIATES of the PGEC (Professional Group on Electronic Computers), and thus receive IRE TRANSACTIONS ON ELECTRONIC COMPUTERS, by payment of $8.50 annually. Apply to IRE Headquarters, 1 East 79 St., New York 21, N. Y.

(IRE members who join PGEC are currently assessed $4.00 per year.)

Professional Societies Approved for Affiliates to PGEC

American Institute of Electrical Engineers
American Management Society
American Mathematical Society
American Physical Society
American Society of Mechanical Engineers
Association for Computing Machinery
Institute of the Aeronautical Sciences
Institution of Electrical Engineers (London)
Instrument Society of America
Mathematical Association of America
National Association of Accountants
National Machine Accountants Association
Operations Research Society of America
Society for Industrial and Applied Mathematics
Society of Automotive Engineers

IRE TRANSACTIONS ON ELECTRONIC COMPUTERS

Volume EC-10 **JUNE, 1961** **Number 2**

Reviews of Books and Papers in the Computer Field

(See front cover for main contents of issue)

(Continued on inside Cover)